MASTER'S THESIS

The achievement of algorithmic accountability through the incorporation of machine learning monitoring methodology

Peeters, T.J.

Award date:
2021

The achievement of algorithmic accountability through the incorporation of machine learning monitoring methodology

Degree:	Open Universiteit, faculteit Bètawetenschappen
	Masteropleiding Business Process Management & IT
Degree program:	Open University of the Netherlands, Faculty Science
	Master of Science Business Process Management & IT
Course:	IM0602 BPMIT Graduation Assignment Preparation IM9806 Business Process Management and IT Graduation Assignment
Student:	Tjadi Jesse Peeters
Identification number:	
Date:	04-07-2021
Thesis supervisor	Vanessa Dirksen
Second reader	Rachelle Bosua
Third assesssor	
Version number:	1
Status:	final version

Abstract

Increasing reliance on algorithms in our daily context is sparking debate regarding the extent to which these algorithms can be held accountable. Opacity as to how algorithmic decisions came to be seems to be the norm and, while principles to which algorithms should adhere have been formulated, these lack proven methods that translate them into practice. Drawing on theories of design-science this research aims to fill that gap by the design of an artefact in the form of a checklist of machine learning monitoring methods that can be used to incorporate algorithmic accountability goals into decision-support systems. A qualitative research approach was taken where, after identifying algorithmic accountability goals from literature, experts in the field of data science were interviewed as to which machine learning monitoring methods could aid in the realisation of these goals. Findings from this stage were later validated using a focus group. Results indicate that the checklist, if embedded in an organisation in a similar strain as security or architectural principles, can aid professionals in the incorporation of algorithmic accountability goals in their decision-support systems.

Key terms

Algorithmic accountability, machine learning, machine learning monitoring, ethical AI

Acknowledgements

It would not have been possible to finish this thesis without the help I have received along the way. For this reason, I would like to extend my gratitude to my thesis supervisor Dr. Vanessa Dirksen for the guidance given during this research project. I believe that without the discussions, feedback and suggestions I would not have been able to produce the same quality of result. I would also like to thank all research participants for their time and insights. My thanks also go to Dr. Rachelle Bosua for co-reading the thesis and (co-)hosting the sessions regarding the research project, which were very interesting and useful. I would like to extend a final thanks to my girlfriend, family and friends.

Summary

This thesis focuses on the manner in which the introduction of machine learning monitoring components can aid in the realisation of algorithmic accountability in decision-support systems. To realise this, the decision was made to create a checklist of machine learning monitoring components that can aid in the realisation of algorithmic accountability goals. After a literature analysis that revealed six algorithmic accountability goals and several machine learning monitoring components which might aid in realising them, two research steps were taken: First four experts in the field of data science were interviewed and confronted with the different algorithmic accountability goals. During these interviews, the respondents hypothesised which components might aid in the realisation of them. After matching results from these interviews with the components identified in literature, an initial design of the checklist was created, which was then evaluated using a focus group consisting of five experts in the field of data science.

The results seem to indicate that the utilisation of the checklist can aid professionals in the incorporation of algorithmic accountability in their decision-support systems. Several suitable machine learning monitoring components have been identified for each algorithmic accountability goal. There are indications that the best strategy for embedding the checklist in an organisation to reach algorithmic accountability is by handling algorithmic accountability in a similar strain as architectural or security principles, where a team within an organisation creates guidelines and methods that should be used and reviews whether projects are adhering to them. Another finding is that professionals indicate that, in certain occasions, GDPR prevent them from fully achieving algorithmic accountability with regards to reproducibility and bias monitoring. Several interesting avenues for future research open up also, where researcher identify the extent to which each of the proposed methods on the checklist impacts algorithmic accountability directly, research what is the effect of different strategies of embedding the checklist in an organisation or look into claims that GDPR are limiting possibilities in achieving algorithmic accountability.

Table of definitions

Term	Meaning
Algorithmic accountability	The degree to which it is possible to hold an algorithm accountable for the predictions it has made (cf. Lepri, Oliver, Letouzé, Pentland, & Vinck, 2018).
Algorithmic lifecycle	The entire process of creating an algorithm from ideation, analysis, design, testing, deployment and monitoring and evaluation (cf. McGregor, Murray, & Ng, 2019).
Data science / machine learning algorithm	A data science / machine learning algorithm is an algorithm that takes a bottom-up approach and extracts rules that it learns from data and uses it for future predictions (cf. McGregor et al., 2019).
Decision-support system	A decision-support system is a system that utilises machine learning algorithms to make predictions and decisions, with little or no human input (cf. Binns, 2018).
Machine learning monitoring (MLM) / Production Model Governance	"The ability to determine the creation path, subsequent usage, and consequent outcomes of an ML model, and the use of this information to accomplish a range of tasks including reproducing and diagnosing problems and enforcing compliance" (Sridhar, Subramanian et al. 2018, p. 351). MLM incorporates system components that interact with data the model is trained on, data the model makes predictions for and all other system components that interact (in)directly with the algorithm (cf. Baylor et al., 2017).
Machine learning monitoring methodology / component	The application of one (or more) machine learning monitoring methods (cf. Baylor et al., 2017).
Metric	A metric is taken to mean a statistical measure that indicates the quality of predictions made by a machine learning model (i.e. the accuracy of a model on a given dataset) (cf. James, Witten, Hastie, & Tibshirani, 2013).
Model	"A machine learning model is the learned program (algorithm) maps inputs to predictions" (Molnar 2020, p. 15).
Prediction	A prediction is what a machine learning model "guesses" a value to be based on other input variables (cf. Molnar, 2020)

Contents

1. Introduction

1.1. Background

Employing machine learning algorithms allows increased efficiency, new insights and reduced costs for decision-making (Lepri, Oliver, Letouzé, Pentland, & Vinck, 2018). Increasing amounts of data and technological advances lead to an increasing reliance on these algorithm types in our daily context, such as employing algorithms to decide whether one should receive a loan or content recommendations on video platforms (De Laat, 2018). While these algorithms have great potential, and might lead to more objective decisions, critics argue that they are prone to enhance discrimination, information and power asymmetry and opacity (Lepri et al., 2018; McGregor, Murray, & Ng, 2019). To address these issues, calls to hold algorithms accountable are increasing. Algorithmic accountability is defined as the degree to which algorithms, or algorithm owners, can be held accountable for predictions algorithms have made (Lepri et al., 2018). It entails the entire framework of monitoring, evaluation and mechanisms allowing an algorithm to be held accountable for predictions made (McGregor et al., 2019).

One of the biggest accelerators in the need for algorithmic accountability has been the move from top-down algorithms, where programmers declare rules explicitly, to bottom-up algorithms - machine learning - which learn rules from data (McGregor et al., 2019). This shift causes algorithms to become more opaque, as rules are not programmed or known exhaustively (De Laat, 2018). To handle this, several frameworks for algorithmic accountability were developed progressively, such as seen in McGregor et al. (2019), but it is noteworthy that so far no comprehensive mature frameworks appear to have been developed. A critique on this field is that, while there have been many initiatives defining principles algorithms should follow, there is a lack of methods that can be incorporated into decision-support systems to implement specified principles (Raji et al., 2020). It has been suggested that this is due to a cultural divide, with engineering on one side and humanities on the other. Legislators, legal scholars, media theorists and ethicists are able to reveal the dangers from increasing algorithm usage, but are unable to articulate them in a way that engineers can incorporate into decision-support systems (Rahwan, 2018).

Concurrent with the field of algorithmic accountability, another field of research called machine learning monitoring (also known as production model governance, production machine learning and model management - referred to from now on as MLM) has been developing. While this field originated with different goals, such as aiding data scientists with their experiment and model management (Vartak et al., 2016), over time the goals of the two fields have grown closer to those of algorithmic accountability. The benefit of the field of MLM being more technical is that research published within it contains specific details that can be incorporated within a decision-support system. MLM acknowledges that the move to machine learning algorithms brings unique governance challenges and has started to share many of the same goals as algorithmic accountability, such as solving management, diagnostic, compliance and regulatory implications of machine learning algorithms in production (Sridhar et al., 2018).

With increased interest in research regarding MLM, researchers such as Sridhar et al. (2018) and Polyzotis, Roy, Whang, and Zinkevich (2018) have proposed MLM frameworks. These generally consist of a set of MLM components, where a MLM methodology / component is the application of one (or several) specific MLM method(s) (Baylor et al., 2017). While these frameworks have been defined,

their methodologies have not yet been incorporated into algorithmic accountability literature, even though they could aid in incorporating defined goals and principles into decision-support systems. Sridhar et al. (2018) note that MLM aids in this regard, as methods provide information regarding the sequence in which an algorithm was created, results and predictions reproducibility and the possibility to audit an algorithm for compliance. This research will draw on literature regarding algorithmic accountability principles and MLM to investigate on achieving algorithmic accountability through incorporating MLM components into decision-support systems.

1.2. Problem statement and research questions

Algorithmic accountability literature has identified issues and abstract solutions regarding the use of algorithms, most of them taking the form of principles that algorithms and their development should follow. Raji et al. (2020) highlight that at least 63 public-private initiatives have developed high-level principles, values or other means to guide the development, deployment and governance of algorithms and AI. The principles are critiqued for being vague, providing little accountability and currently lack proven methods allowing them to be incorporated into decision-support systems (Greene, Hoffmann, & Stark, 2019). Without any further operationalisation of the principles, they become the operational goals within organizations (Raji et al., 2020), but translating such vague goals into something implementable in a system is a struggle for professionals trying to achieve algorithmic accountability (Rahwan, 2018).

Within the data science community, increased adoption of machine learning in systems has encouraged calls for increased machine learning monitoring (Sculley et al., 2015). MLM could be seen as a necessary layer for enforcing algorithmic accountability for machine learning. Meaning, to be able to enforce algorithmic accountability sufficiently, necessary MLM practices need to be established. An example is that one must be able to gain insight into how predictions from an algorithm came to be to hold an algorithm accountable for them (De Laat, 2018). An incorporation of MLM methodology into current algorithmic accountability goals can help defining these principles in manners allowing professionals to incorporate them into their decision-support systems. To aid in this matter, the central question of this research is:

> *"How can a machine learning monitoring methodology be incorporated into decision-support systems so as to achieve algorithmic accountability?"*

To answer this central question, two sub-questions are posed:

1. *Which machine learning monitoring components fit current algorithmic accountability goals?*
2. *How could an artefact incorporating these components be embedded in an organisation so as to achieve algorithmic accountability in practice?*

1.3. Research objective

The research objective is assessing how MLM methodology can be incorporated into decision-support systems so as to achieve algorithmic accountability. To answer this question, a design science research approach will be applied to develop a checklist of MLM components that professionals can employ to incorporate algorithmic accountability goals into their decision-support systems. A decision was made for a checklist as current principles are lacking methods that allow professionals to incorporate them into their decision-support systems. By employing checklists, professionals can determine suitable methods for a specific use-case. This is in line with literature regarding design-science, where

technological rules are listed as IT-artefacts in an abstract form (Baskerville, Baiyere, Gregor, Hevner, & Rossi, 2018), and research showing guidelines to be a common type of artefact using in design science for information systems (Offermann, Blom, Schönherr, & Bub, 2010).

1.4. Motivation/relevance

Current algorithmic accountability frameworks and their principles for machine learning models lack a set of methods that can be incorporated into decision-support systems (Rahwan, 2018; Raji et al., 2020). Without these, professionals are unable to ensure they are incorporating algorithmic accountability sufficiently. Incorporating a MLM methodology into current algorithmic accountability principles can aid this, by providing a list of methods that professionals can implement into decision-support systems. Using this, professionals can improve the ethics of their predictive models for decision-making, enhance transparency of algorithms used and ensure that data used for input is suitable. Additionally, the research aims at bridging the gap between the humanities posing problems in terms which cannot be matched by solutions from technical literature - as identified in Rahwan (2018) – by utilizing both literature fields.

1.5. Main lines of approach

The research method is mainly exploratory and follows a design science research approach. The decision was made to utilise design science, as it is a research paradigm aiming to solve problems through the creation of artefacts (Hevner, March, Park, & Ram, 2004). This seemed especially appropriate for the aim of producing a checklist that can be used to incorporate algorithmic accountability into decision-support systems. The research is organised along the framework of three design cycles highlighted in Hevner (2007).

1.6. Overview

The rest of this paper is structured as follows: Chapter 2 discusses the theoretical background and literature analysis carried out. Chapter 3 discusses the research methodology applied to inspire the design and evaluate the artefact in the environment. Finally, Chapter 4 and 5 discuss the results and the conclusions of the research, respectively.

2. Literature and theoretical background

This section discusses the literature relevant for answering the research question and how it was analysed. This is part of the design science rigor cycle, where grounding of the artefact in the knowledge base should be ensured. Section 2.1 discusses the approach taken to analyse the literature, followed by Section 2.2, which starts with discussing relevant goals within algorithmic accountability and closes with discussing current MLM components. The result of the literature analysis will provide two tables that are used to create an initial design of the desired artefact that has grounding in the current knowledge base.

2.1. Literature review approach

The literature review is conducted to first gain insight into which methodologies and frameworks for MLM currently exist, which components they consist of and which parts are fit for incorporation into algorithmic accountability. To achieve this, a literature review of current MLM practices will be done alongside a review of current algorithmic accountability frameworks.

The literature research follows the grounded theory approach outlined in Wolfswinkel (2011). It is a recommended way to do a rigorous and theoretically relevant analysis of a topic where data has the form of published articles (Wolfswinkel, 2011). The first step in this process is defining the literature subset of literature being analysed. Thus, the criteria for inclusion and exclusion were defined. A first criterion was that an article had to be related to frameworks regarding algorithmic accountability or be regarding MLM methodology. Another criterion was that research had to be a journal or conference paper published in the last 3 years. Relevant papers published before this were later identified through backward citation tracking.

The second and third steps in Wolfswinkel (2011) are search and select, respectively. To start, relevant articles are retrieved from a database using an appropriate search term. In this research, articles were sourced from the library of the Open University. After applying the search terms displayed in Table 2.1, the articles for analysis were selected, which entailed scanning titles and abstracts of articles regarding their relevance for the criterion. Forward and backward citation tracking was used to expand the dataset.

Table 2.1: Literature subsets and search terms

Literature subset	Search term
Algorithmic accountability	(algorithmic accountability OR algorithmic transparency) AND (((definition OR explanation OR interpretation) OR (framework OR structure)) OR practices)
Machine learning monitoring	(machine learning OR production model OR algorithm OR model) AND (governance OR monitoring)

The search term regarding algorithmic accountability resulted in a total of 397 results on 29-09-2020. Based on the above-listed criteria and the scanning of the abstract and titles, this set of papers was reduced to 19. By backward and forward citation tracking this was then extended to 25 papers in total.

The query regarding MLM resulted in a substantial set of 122,053 results; a scan showed the greater part was not regarding the topic of interest. After failing to adjust the search term successfully, most likely due to topics of high interest such as machine learning and model being in it, an alternative search strategy was used: search results were filtered to relevant technical conferences where one could expect relevant papers to be discussed. Selected conferences were Neural Information

Processing Systems (NIPS), Special Interest Group on Data Management (SIGMOD) and USENIX. These were selected as currently MLM methodology seems to be most prominently discussed in technical areas and the level of detail that can aid algorithmic accountability by allowing incorporation into decision-support systems would most likely also be found. The search delivered 5 relevant papers. Forward and backward citation tracking of these articles expanded the set to 19.

The fourth step in Wolfswinkel (2011) is analysis using open, axial and selective coding. In this case, an initial distinction made between two literature subsets, namely algorithmic accountability and MLM. During the analysis step, further papers were still removed from the literature subset, as they were found to not meet defined criteria, giving us a final subset of 13 and 11 papers regarding algorithmic accountability and MLM, respectively. As the literature research goal was to find the most suitable MLM components to incorporate algorithmic accountability goals into decision-support systems, subsets were coded differently. The coding and analysis process used can be decomposed into two distinct steps. It must be noted that, as appears to be the case for all literature research, steps were not followed as linearly as one might assume based on the description. Often, to ensure a good fit between the segments of literature, information of one step informed going back to adjust decisions made earlier.

First of all, the MLM subset was analysed. The goal was to distil available MLM components from the literature, defining a MLM component as the utilisation of a MLM methodology that can be incorporated into a system. To do this, first open coding was applied, where each concept identified was a MLM component. Afterwards, a list of all found components was analysed and axial and selective coding applied, analysing relations between concepts and identifying higher-order (sub)categories to which these components belonged and deduplicating identical concepts. Concepts and (sub)categories were analysed based on their appropriateness for the research question. The output of this coding exercise was a table of categories, subcategories and concepts for MLM components.

The second step entailed the algorithmic accountability subset coding. Given the goal of the literature analysis is to find the manner by which MLM can aid in the incorporation of algorithmic accountability into decision-support systems, it was deemed appropriate to distil the algorithmic accountability goals from the literature subset. This is the extent to which MLM components can aid in the achievement of these goals can be seen as their appropriateness in their incorporation. For this, every excerpt in the literature subset defining a principle that an algorithm should follow, a goal for algorithmic accountability, a desirable property an algorithm should have or an issue with current algorithms was coded. During this open coding step these excerpts were analysed to distil the inherent goal from the statement. After this, axial and selective coding could be done to identify categories of goals, deduplicate redundant goals and select the ones relevant to the research questions. The output of this coding step was a table containing categories and concepts for the goals of algorithmic accountability.

2.2. Literature background - results and conclusions

This section discusses the literature analysis results. It will start with a discussion of algorithmic accountability followed by MLM.

2.2.1. Algorithmic accountability

During the open, axial and selective coding as described in (Wolfswinkel, 2011), several categories, sub-categories and concepts of goals were defined within algorithmic accountability. The overview is listed in Table 2.2 and discussed below after an introduction to algorithmic accountability.

Weakening accountability of systems due to our increasingly computerized society is not a phenomenon. For example, the responsibility diffusion trend in systems created by many was signalled decades ago (De Laat, 2018). A new trend is a growing reliance on algorithmic decision-making, where, while we increasingly rely on systems utilising algorithmic decision-making, it is unclear how these algorithms can be held accountable (De Laat, 2018). Accountability for a system can be taken to mean the responsibility for its behaviour and impacts (Raji et al., 2020). Following in this regard, algorithmic accountability can be defined as the degree to which an algorithm, or the owner of an algorithm, can be held accountable for predictions made by the algorithm (Lepri, Oliver et al. 2018). Although as entities not having any legal status, algorithms themselves cannot be held accountable, organizations employing them can be through the use of governance structures and implementation of checks and balances within processes (Raji et al., 2020).

The most common approach encountered in the literature to achieve algorithmic accountability - or ethical AI – is by producing high-level principles, such as transparency and fairness, and values that algorithms should adhere to (B. Mittelstadt, 2019). While these approaches are an understandable starting point, they have their flaws and are critiqued for being vague and lacking proven methods that can be incorporated into decision-support systems (Rahwan, 2018; Raji et al., 2020). B. Mittelstadt (2019) compares algorithmic accountability to the field of medicine for which methods matching their principles are available.

While they might lack proven methods that can be implemented, these approaches have highlighted many goals regarding what attributes algorithms should have and which checks and balances should be in place (B. Mittelstadt, 2019). The categories of these goals that emerged during the literature research are shown in Table 2.2. The following goals to achieve algorithmic accountability were identified: systems utilising an algorithm need to provide information that allows them to be audited, analysis should be done to monitor whether an algorithm is performing accurately and as expected, deeper analysis should be done to monitor and prevent unintended effects, an algorithm cannot contain any biases, a human should be kept in the loop and finally that algorithms should have mechanisms in place making them interpretable.

Table 2.2 Categories and concepts of algorithmic accountability goals

Categories of Accountability: Concepts		
Category	**Concept**	**Literature**
Algorithmic analysis	Monitoring and preventing unintended effects	(Rahwan, 2018)
Algorithmic analysis	Measuring whether an algorithm is performing accurately and as expected	(De Laat, 2018; McGregor et al., 2019; Rahwan, 2018; Shin & Park, 2019)
Algorithmic analysis	Algorithms cannot contain any bias	(De Laat, 2018; McGregor et al., 2019; B. D. Mittelstadt, Allo, Taddeo, Wachter, & Floridi, 2016; Zarsky, 2016)
Human-in-the-loop	Human-in-the-loop	(Danaher et al., 2017; McGregor et al., 2019; B. D. Mittelstadt et al., 2016)
Outcomes	Interpretability	(Binns, 2018; Danaher et al., 2017; De Laat, 2018; Lepri et al., 2018; McGregor et al., 2019; B. D. Mittelstadt et al., 2016; Shin & Park, 2019; Zarsky, 2016)
Overall system / model insight	The systems utilising an algorithm must be auditable	(Lepri et al., 2018; McGregor et al., 2019; Raji et al., 2020)

Understanding the possibility of unintended consequences is important when addressing algorithmic accountability, as engineers and managers should have checks and balances monitoring them (Shin & Park, 2019). This is relevant, as a designer of algorithms can only be held accountable when there is a certain level of control over decisions made, implying a responsibility to monitor algorithms (B. D. Mittelstadt et al., 2016). In this regard, a lack of interpretability of machine learning systems threatens the possibility of being held accountable for their systems actions, as it is unclear how those actions came to be (Binns, 2018). Determining whether a problem is a bug or systemic failure when algorithms are opaque and hard to interpret is difficult (B. D. Mittelstadt et al., 2016). Due to the interrelated nature of machine learning, the responsibility to monitor should cover the full algorithmic lifecycle and cover topics such as bias and discrimination (De Laat, 2018). For the possibility of monitoring the entire algorithmic lifecycle, the process which produced the predictions and algorithms must, to a certain extent, be reproducible and auditable (Shin & Park, 2019).

It is this reproducibility of results to give insights into predictions and interpretability that generally go hand in hand. Based on public reason theory, Binns (2018) argues that the system utilizing algorithms should have a justification that has epistemic and normative standards which are 'acceptable to all reasonable people'. A common example is of a credit company giving a justification for its prediction justifying its modelling approach, scores of the model, anti-discriminatory measures taken and the influence of input data. Similar arguments for accountability where a connection regarding the full lifecycle of an algorithm, regarding the connection of input data, data transformation, to algorithm, need to be given is made is mimicked in (De Laat, 2018; Lepri et al., 2018; Shin & Park, 2019).

Another way of monitoring algorithms is by keeping a human-in-the-loop (B. D. Mittelstadt et al., 2016). A human-in-the-loop serves two purposes: the spotting of misbehaviour by the system and providing an accountable entity in case of misbehaviour (Rahwan, 2018). Yet another way to monitor algorithms can be to do analysis to control whether algorithms are working as expected (Rahwan, 2018; Shin & Park, 2019). Besides this constant monitoring, regular audits and more thorough analysis can be done to look for any unintended consequences of algorithms (Rahwan, 2018; Shin & Park, 2019).

2.2.2. Machine learning monitoring mechanisms

Following the algorithmic accountability goals analysis, a MLM literature subset was analysed to identify MLM components that aid in the incorporation of algorithmic accountability into decision-support systems. Below a short introduction is given into MLM, after which identified component categories are discussed. During analysis several categories emerged, namely model validation, assessing the appropriateness of data and reproducibility & auditability of results. An overview of these categories and concepts is shown in Table 2.3. A full matching of each concept and literature discussing it can be found in Appendix 1.

A MLM system deals - next to code directly related to algorithms - with all other functions related to the reliable deployment of machine learning into systems (Sculley et al., 2015). Its goal is to achieve the ability to determine the creation, usage and outcomes of a machine learning algorithm (Sridhar et al., 2018). For this, many different infrastructure parts are needed, as displayed below, in Figure 2.1. The authors Sculley et al. (2015) highlight that the black-box – code directly related to machine learning algorithms – is small compared to other infrastructure which is vast and complex. Creating and deploying machine learning models reliably is complex and requires the orchestration of many different components, of which only some interact directly with the model (Baylor et al., 2017). The interaction of all these components can be seen as MLM (Sridhar et al., 2018).

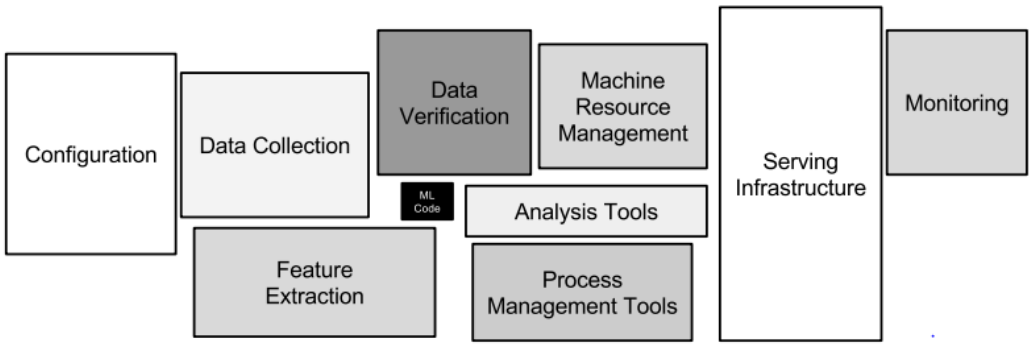

Figure 2.1: infrastructure components needed to reliable deploy algorithms (Sculley et al., 2015)

The category of model validation entails MLM components used to validate whether models are (still) working as intended, without unforeseen behaviour (Baylor et al., 2017; Sridhar et al., 2018). As these models are trained on a specific dataset from which they extricate patterns utilised in predictions, the appropriateness of this data is essential to its performance (Polyzotis et al., 2018; Sculley et al., 2015; Sridhar et al., 2018), hence the introduction of this subset of MLM components as the second category. A third category is the reproducibility and auditability of machine learning results. Due to the lack of independent components within machine learning systems, it is vital that results can be reproduced to identify the source of problems and audited to control regulation compliance (Schelter, Boese, Kirschnick, Klein, & Seufert, 2017). A common example of the need for reproducibility and auditability is the abbreviation CACE (Changing Anything Changes Everything), which implies that a small change in machine learning system configuration might change results completely (Sculley et al., 2015).

Table 2.3 Categories, sub-categories and concepts of MLM components

Machine learning monitoring mechanisms			
Category	**Description**	**Concepts within category**	**Literature**
Model validation	This category exists of components used for the validation of whether models are (still) working as intended, without any unforeseen consequences.	Human in the loop, Sanity Checks, Metrics on entire dataset, Metrics on data slices, Visual exploration of predictions, Detection of changes in prediction behaviour, Fairness Metrics, Alerts that notify an admin.	(Arrieta et al., 2020; Baylor et al., 2017; Crankshaw et al., 2015; Kahng, Fang, & Chau, 2016; Miao, Li, Davis, & Deshpande, 2017; Polyzotis et al., 2018; Sculley et al., 2015; Sridhar et al., 2018; Vartak et al., 2016)
Assessing the appropriateness of data	The category entails components used as checks and balances of the data on which the model is trained (or scheduled to be trained).	Data validation, Automated Model Training, Controlling for training-serving skew, Statistics for data change detection	(Baylor et al., 2017; Crankshaw et al., 2015; Miao et al., 2017; Polyzotis et al., 2018; Sculley et al., 2015; Sridhar et al., 2018; Vartak et al.)
Reproducibility & auditability of results	The category exists of components that are used to make predictions and results reproducible and auditable.	Configuration Framework, Metadata management, Versioned copies, Lineage tracking and provenance.	(Baylor et al., 2017; Miao et al., 2017; Polyzotis et al., 2018; Schelter, Boese, Kirschnick, Klein, & Seufert, 2017; Sculley et al., 2015; Sridhar et al., 2018; Vartak et al., 2016)

i) Model validation

A category of solutions that emerged from literature was model validation. This category entails strategies to validate whether a model is (still) working as intended and that there is minimal unforeseen behaviour. A list of concepts within this identified category is shown in Table 2.2. The sub-categories identified were metrics, human and predictions.

The first sub-category, metrics, entails the tracking of metrics indicating model performance. A common strategy for this is calculation of metrics on the entire dataset on which a model is trained. Engineers should be able to define metrics and thresholds indicating performance problems which are computed when their models are trained (Baylor et al., 2017). As machine learning systems increase in complexity, it is often needed to analyse training data subsets to analyse model behaviour, as a model can have good performance metrics for the entire dataset, but still have sub-groups for which it performs poorly (Kahng et al., 2016). Defining metrics monitored for specific data slices, as this can be used to identify underperforming areas, should be possible (Baylor et al., 2017; Polyzotis et al., 2018).

The second sub-category is predictions. Frequently, the behaviour of a model might change on new data (Baylor et al., 2017; Sculley et al., 2015), indicating that metrics also need to be computed for their predictions. Analysis controlling for changes in prediction behaviour is important, as these might indicate differences in data for which the predictions are made, meaning that the distribution of predicted labels should generally be the same as the distribution of observed labels in the training dataset (Sculley et al., 2015). Predictions made by a model should also be fair. Fairness metrics can be used to limit the bias amount that a model can have in its predictions (Arrieta et al., 2020).

The final sub-category is human. Many steps within a machine learning system can happen automatically, but generally, it is advised to have a human-in-the-loop make certain decisions and analyses regarding model deployment. Next to the benefit of extra validation of algorithms, the logs of these decisions can be used during audits for compliance with regulations (Sridhar et al., 2018). A human-in-the-loop can take the form of a human taking decisions whether to deploy a model as

discussed in Sridhar et al. (2018), but also a human investigating alerts when certain thresholds indicate anomalies during any stage of a model lifecycle (Baylor et al., 2017).

In conclusion, model validation entails a category of MLM components that can be utilised to validate whether a model performs well on metrics, its predictions are as expected and the incorporation of a human to do validations during the process.

ii) Assessing the appropriateness of data

As noted in the introduction, much of the opacity in algorithmic accountability is due to the bottom-up, data-nature of machine learning. The same is noted in Polyzotis et al. (2018) where it is said that the accuracy of a machine learning model is deeply tied to the data it is trained on. Sridhar et al. (2018) note that the data-dependent nature of machine learning causes small changes to have unintended consequences in predictions. To prevent data-related issues from affecting a machine learning model in production, several measures were proposed, which are labelled under the category assessing the appropriateness of data. The full list of concepts for this identified category is listed in Table 2.2; the sub-categories are data changes and data validation.

Data validation is a category of components ensuring that no predictions are made on invalid data. Initial checks that should be done on any incoming data are to see whether data given to a model has the correct shape for predictions. These are called sanity checks (Baylor et al., 2017; Polyzotis et al., 2018). For this reason, descriptive statistics on every feature of the model have to be computed. These are then used to ensure that data errors are found and amended, preventing errors from flowing downstream and invalid models from reaching production (Baylor et al., 2017).

Data changes, this category relates to the common issue of real-world patterns changing from patterns the model extracted from training data. Controls checking whether patterns in training data and thus extracted by the model are still valid for new data have to be in place. In cases where this is no longer the case and models no longer represent the world accurately, one speaks of concept drift (Baylor et al., 2017; Crankshaw et al., 2015). When any deviations exist between training and test data, the model will not perform consistently when deployed. For this reason new data should be compared to training data and any changes should be detected and reported (Baylor et al., 2017; Sculley et al., 2015). Changes in data can be found using statistical measures such as homogeneity test, analysis of variance, time series analysis, or change detection (Polyzotis et al., 2018). This changing nature of data and the world displays the need for a certain degree of automation in model training. Models may need to be retrained on new data as old models may no longer represent the current state of the world (Crankshaw et al., 2015).

To summarise, the category appropriateness of data consists of MLM components allowing data validation, controlling whether patterns extracted from earlier data are still valid and ensuring that updated data is used when needed.

iii) Reproducibility & auditability of results

The final category that emerged from literature is reproducibility & auditability. It entails everything necessary to produce experiments and predictions reproducibility and the possibility of auditing them for issues and compliance to regulations. All machine learning components under this identified category are listed in Table 2.1.

Reproducibility is a key issue in machine learning systems, as the final model in a machine learning system generally was produced in an iterative manner, where many possible solutions are tested before a solution meeting the criteria is found (Vartak et al., 2016). Another consideration is that

machine learning experiments are highly dependent on many aspects, including data versions, parameters used, metrics that were optimized and which algorithms were compared. These experiments are often incomparable. For achieving reproducibility and repeatability, metadata must be understood and provenance of artefacts produced by ML experiments should be known. Using metadata analysis one can answer questions such as 'Who created the model at what time?' and lineage allows questions in a similar strain to 'What dataset was the model derived from?'. Answering these questions is essential for comparability and repeatability of ML experiments (Baylor et al., 2017; Miao et al., 2017; Schelter et al., 2017; Vartak et al., 2016). In this regard, provenance and lineage should allow the exact sequence of events (data, training logs, code, pipelines and human approvals) that led to the conclusion (Baylor et al., 2017). To allow this, metadata artefacts that should be stored are version lineages of data, scripts, results, data provenance, workflow metadata and dependencies (Baylor et al., 2017; Miao et al., 2017; Schelter et al., 2017; Vartak et al., 2016). Furthermore, metrics regarding model performances should be saved for all model versions, as this allows easier diagnosis of regression in model quality (Crankshaw et al., 2015). Here, lineage information can be useful, as it aids analysis of the training sample, which parameters were used and other essential information when comparing different model versions for improvement and validation (Miao et al., 2017).

In conclusion, the category of reproducibility & auditability consists of MLM components required to reproduce results and increase system auditability.

2.2.3. Follow-up research

This initial part of the research has produced two tables containing algorithmic accountability goals and MLM components, respectively. The concepts in both tables provided theoretical grounding in the knowledge base. With this grounding, algorithmic accountability goals were utilised to structure the interviews and confirm which of the found MLM components are suitable to aid in their realisation. These results were then developed to inspire an initial design of the artefact in the form of a checklist of MLM components that can be incorporated into decision-support systems so as to achieve algorithmic accountability that was then further evaluated using a focus group. The methodology for these research steps is discussed in the next chapter.

3. Methodology

This section of the research describes the research design in detail. It will start with a description of research activities, after which a description of data collection activities and analysis are given. Finally, the chapter ends with a reflection on quality criteria for qualitative research.

3.1. Research activities

The design science paradigm is suited for executing this research, as the goal of is to produce an artefact in the form of a checklist of MLM components that professionals can use to incorporate algorithmic accountability principles within their decision-support systems. It stems from engineering and is fundamentally a problem-solving paradigm (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007). Hevner et al. (2004) contrast the behavioural-science paradigm, which seeks to create and substantiate theories explaining phenomena surrounding information systems, with design-science for IS research, that aims to provide solutions to existing problems through the creation of artefacts. During this research, a qualitative approach is used to evaluate the artefact. The interpretive nature of qualitative research, where the researchers need to make sense of what they are studying suits the exploratory nature of the research and allows the flexibility often needed during this type of research (Thornhill, Saunders, & Lewis, 2009).

The framework for the organisation of design science using three-cycles from Hevner (2007) is displayed in Figure 3.1 with research activities taken during each cycle added. To support initial grounding in the knowledge base, the table of goals discovered during literature analysis provided structure to the interviews in step 2. During this step, to answer sub-question one, respondents listed potential components that could aid in realisation of the goals, which were then later compared to identified components from the literature research. For answering sub-question two a discussion into the requirements for the artefact and its evaluation was done. Step 3 then consisted of consolidating findings and creating an initial design of the artefact, which could later be validated and evaluated during the focus group in step 4, where the components were discussed once more and how the artefact should be embedded in organisations was discussed. The metrics against which the artefact was evaluated were efficacy and usefulness from Mijač (2019). These are listed as the most commonly adopted in the evaluation of design-science artefacts in information systems. Table 3.1 provides an overview of the metrics and their descriptions from Mijač (2019), followed by a translation of the metric for the research question.

Table 3.1 Top 2 metrics used in the evaluation of software engineering artefacts from (Mijač, 2019)

Criteria	Description	Translation to research question
Efficacy	The degree to which the artefact achieves its goal considered narrowly, without addressing situational concerns.	The degree to which the artefact allows professionals to incorporate algorithmic accountability principles into decision-support systems.
Usefulness	The degree to which the artefact positively impacts the task performance of individuals.	The degree to which the artefact positively allows professionals to incorporate algorithmic accountability principles through the implementation of MLM components into their decision-support systems.

After incorporating the findings of the focus group, in step 5 a validated checklist was added to the existing knowledge base. As can be seen in Figure 3.1., this research will consist of a single iteration through all three-cycles, after which future research can continue with new iterations through the cycles. This is discussed more elaborately in the directions for future research in Chapter 5.

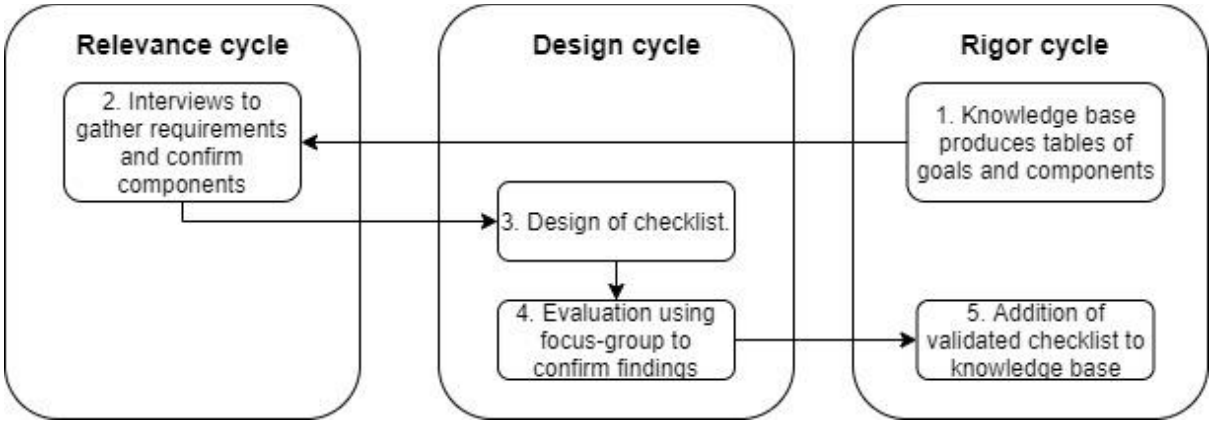

Figure 3.1: Overview of steps taken in each of the three design science cycles from Hevner (2007)

3.2. Data collection methods

3.2.1. Semi-structured interviews

The initial research step where requirements for the artefact were gathered and a confirmation of the components contributing to the realisation of each goal was done using semi-structured interviews. Semi-structured interviews were used, as this method allows for the discussion of a list of themes and key questions, while maintaining flexibility for new topics to arise (Thornhill et al., 2009). This was especially suited for the confirmation of the components, as the interview could take the structure of the goals identified during the literature review, where respondents could then openly hypothesise what components are most suitable to reach this goal. During analysis, these components could then be compared with the ones from the literature. In this way, unwanted influence by the researcher as to which components are suitable was prevented. Participants for the interviews at this stage were selected based on their expertise in data science and it being reasonable for them to receive a request to incorporate algorithmic accountability principles into decision-support systems. This led to the selection of data science professionals. To adhere to the recommendation from Thornhill et al. (2009), that the generalisability of research can be improved by the selection of a wide cross-section of participants, it was ensured that participants from multiple organizations were interviewed. To prevent any bias in the interpretation of the interview, the results gathered were sent to interviewees for validation. The interviews followed a common structure, where part one focused on answering which MLM components fit which algorithmic accountability goals and parts two and three focused on what requirements an artefact incorporating these should have to be embedded in organisations:

1. Discussion of each algorithmic accountability goal and MLM components contributing to its realisation
2. Discussion of requirements for the artefact and its evaluation on usefulness and efficacy
3. Other comments regarding artefact

Besides this structure, an interview protocol was created (Appendix 2).

3.2.2. Focus group

The final design of the checklist was evaluated during a focus group. Focus groups are group interviews with a clearly defined topic focusing upon discussion between participants (Thornhill, Saunders et al. 2009). Specifically, interaction between participants is being researched in a focus group (Morgan, 1996). It is a valuable tool in combination with semi-structured interviews and can aid in confirmation of earlier results and reaching theoretical saturation (Kitzinger, 1994). In this case, participants are professionals for whom it is reasonable to expect that they receive requests to incorporate algorithmic accountability in a decision-support system, leading to the selection of data scientists and data engineers that have worked on incorporating MLM in their organisation. During this focus group, the artefact was presented and earlier findings were discussed. To answer how an artefact incorporating these components should be embedded in an organisation in practice a discussion was held regarding this topic. This led to the focus group having the following structure:

1. Discussion of each proposed algorithmic accountability goal and the MLM components proposed to realise it
2. Discussion regarding how the artefact should be embedded in organisations
3. Evaluation of artefact on efficacy and usefulness
4. Other comments regarding artefact

Next to this structure, a focus group protocol was created (Appendix 3).

3.2.3. Participants in the research

An overview of the research participants is given below in Table 3.2.

Table 3.2 Overview of research participants

Respondent	Position	Experience	Years of experience in data science
INTRESP1	Data Science Consultant	Master's degree in Marketing Intelligence. Currently working as a Data Science Consultant.	4
INTRESP2	Data Scientist / CEO	Master's degree in Business Analytics. Former Lead Data Scientist of a data science team at an aviation firm. Founder of a company that develops software for MLM.	6
INTRESP3	Senior Machine Learning Engineer	PhD in Astrophysics. Currently employed as a Senior Machine Learning Engineer with focus on MLM and responsible AI at a bank. In the process of starting a foundation that provides information regarding ethical machine learning.	5
INTRESP4	Machine Learning Engineer	Master's degree in Artificial Intelligence. Former Researcher at an university in Sweden. Currently employed as a Machine Learning Engineer for a software company.	5
FOCRESP2	Data Science Lead	Master's degree in Econometrics. Currently employed as data science lead for an FMCG firm.	17
FOCRESP3	Data Scientist	Master's degree in Data Science. Currently focusses on building marketing analytics algorithms as a data science consultant at an FMCG firm.	2
FOCRESP4	Data Scientist	Master's degree in Artificial Intelligence. Currently focusses on building a data science platform at an FMCG firm.	8
FOCRESP5	Senior Data Engineer	Master's degree in Software Engineering. Currently focusses on building a data science platform at an FMCG firm.	5
FOCRESP6	Data Engineer	Master's degree in Business Informatics. Currently focusses on building a cloud infrastructure for a data science platform at an FMCG firm.	2

3.3. Data analysis

The data collected during interviews was in the form of recordings (with consent), that were transcribed and coded using the criteria from Table 3.1. To best answer which MLM components fit which algorithmic accountability goals, a combination of top-down and open coding was used. Top-down coding is characterized by already knowing the topics of interest that will be coded. Open coding extracts codes and topics of interest from the data (Thornhill et al., 2009). The preliminary stage of top-down coding conducted is for validation of items on the checklist of the artefact. This is completed by the coding of the respondents' answers following the coding tree in Figure 3.2.

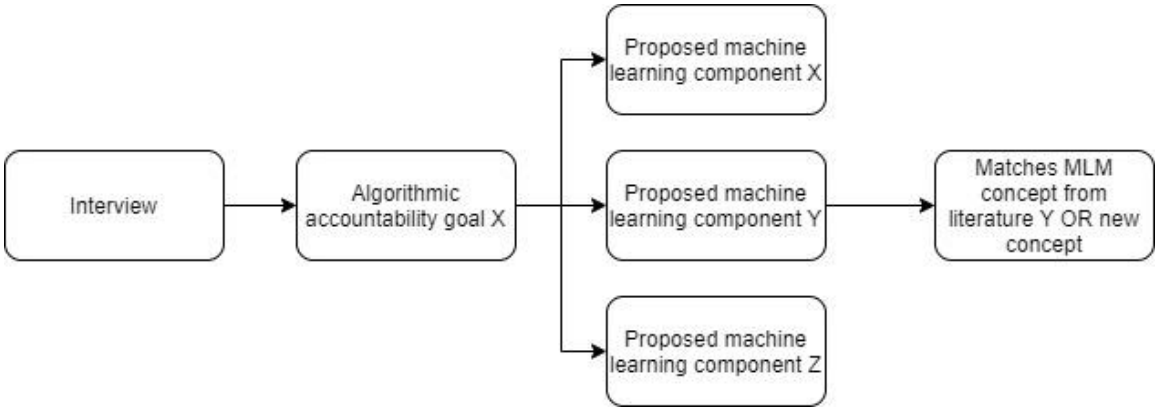

Figure 3.2: Coding tree for evaluation of items on the checklist of MLM components

To answer how an artefact incorporating MLM components can be best embedded in an organisation in practice, a combination of top-down and open coding was also used. An initial stage of open coding was used, where all statements regarding improvements or adjustments regarding the artefact were coded. After this, these topics were combined into improvement and requirement categories, which could then be related to whether they would improve efficacy or usefulness of the artefact. The coding tree for this is shown in Figure 3.3.

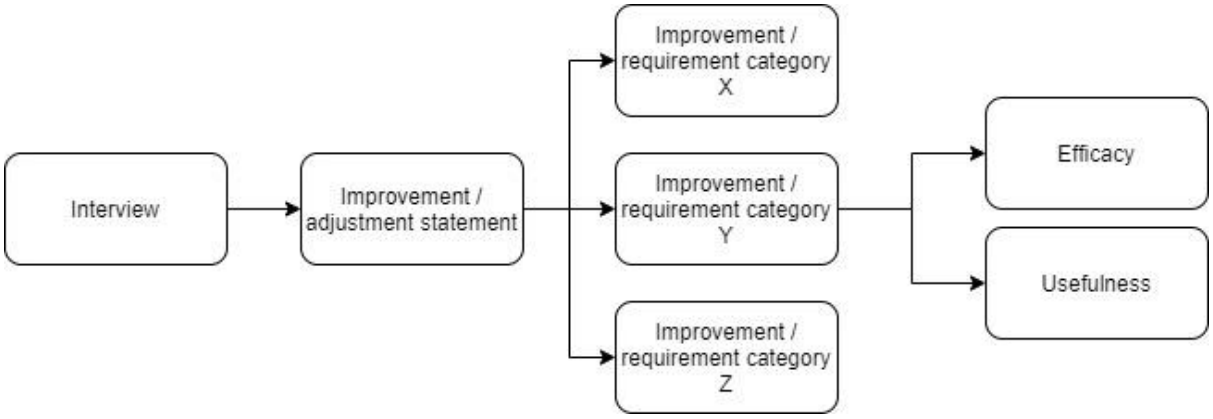

Figure 3.3: Coding tree for requirements gathering and evaluation of artefact

The data during the focus group was collected through recording and was analysed using coding. The coding follows the same structure as the interviews, utilising the coding trees displayed in figures 3.2 and 3.3 – with the addition that ways of embedding the artefact in organisations were also coded. The transcriptions for both the interviews and the focus group will be available on request for any future researchers.

3.4. Reflection on quality criteria and ethical aspects

This section reflects upon the research design's impact on quality criteria for qualitative research. It will begin with a discussion of trustworthiness, followed by a reflection on the criteria authenticity and ethical aspects of the research design.

3.4.1. Trustworthiness

The criteria trustworthiness parallels the rigor criteria of quantitative research and consists of credibility, transferability, dependability and confirmability (Lincoln & Guba, 1986). Credibility refers to the extent to which results are accurate (Lincoln & Guba, 1986). While they might produce lower

external validity, qualitative measures are extremely appropriate for maximizing credibility (Thornhill et al., 2009). To maximize credibility, method triangulation, meaning that the initial interviews informed the focus group structure, allowing more flexibility and adjustments to better fit the data being gathered for answering the research question was used. It also provided analysis of multiple data sources to confirm validity of results (Thornhill et al., 2009). Another measure to improve credibility was use of participant validation, where results were later shared with participants so they could confirm or amend them. Unfortunately – due to the structure of the research being a thesis – it was impossible to use multiple researchers, even though this can aid credibility (Lincoln & Guba, 1986). Another factor impacting credibility negatively is the limited time available to conduct research. Consequently, the research design utilising design science looked somewhat different from what it would look in an ideal setting. The three-cycle view by (Hevner, 2007) displays the initial need to retrieve requirements from the environment, in the case of this research this had to coincide with an initial evaluation. The artefact is inspired by the literature, but at that point, no empirical research was done yet. Moreover, no field-testing of the artefact, in this case the actual use of the artefact in the design of a decision-support system, was possible due to limited time available. Instead, the final evaluation of the artefact consisted of a focus group. Ideally, design science research would also contain multiple iterations of all research cycles, but unfortunately, this was impossible due to time constraints.

Transferability is the extent to which research results are applicable in different contexts (Lincoln & Guba, 1986). Usage of semi-structured interviews and a focus group make it difficult for a research to be reproducible, as results are dependent on the time and place where they are taken (Thornhill et al., 2009). To alleviate this effect, the interviews and focus group had a common thematic structure given by the selected metrics to evaluate the artefact and results of the literature analysis. Additionally, the interviews and focus group were recorded, transcribed and analysed by coding. Ensuring the reproducibility of the analysis of the data. Other measures for ensuring reproducibility are literature analysis to distil the artefact from an existing literature subset, a description of research design and selection of participants from multiple organizations. The participants of the research were also carefully selected, ensuring that they could answer questions posed.

Dependability and confirmability refer to the consistency of research findings and the degree to which they are confirmed by other researchers (Lincoln & Guba, 1986). An attempt to ensure dependability was made through method triangulation in the research design. In this case, the confirmability was difficult to measure, as at the time the researcher is unaware of any other existing checklist which intends to incorporate algorithmic accountability in decision-support systems using MLM components. An argument could be made that the original artefact, as it is inspired by the literature, enhances confirmability.

3.4.2. Authenticity and ethical aspects

Ethical research means using appropriate behaviour standards to guide conduct regarding the subjects of your research (Thornhill et al., 2009). The quality criteria authenticity relates to this and addresses the impact on the group being researched (Lincoln & Guba, 1986). Following high ethical standards was the aim during this research. All data were collected with consent of participants, who were later given a chance to withdraw from the research and found results were validated with them. The found data was anonymized and securely saved where there were no risks of improper access being gained.

4. Results

This chapter discusses the findings following the execution of the research as described in Chapter 3. Firstly, the interviews focused on what requirements the checklist should have and which components can aid in the algorithmic accountability goals realisation. This output led to an initial analysis (Appendix 4) and an initial checklist design. The next iteration of research consisted of the focus group where the emphasis was on how to embed such a checklist in an organisation and validating the components identified during the interviews. The analysis of this research step can also be found in Appendix 4. The findings below discuss the results of these two iterations of research and analysis and are structured according to the themes of algorithmic accountability goals identified during the literature analysis.

4.1. Algorithmic analysis

4.1.1. Performance

For reaching the goal of **measuring whether an algorithm is performing accurately and as expected,** all interview respondents agreed that performance monitoring of machine learning algorithms is needed to ensure an algorithm is working as expected. The approach mentioned most often for achieving this is the monitoring of metrics. One drawback identified for this approach is that one must define thresholds that indicate problems, which are often subjective and to a certain extent use-case dependent. This measuring of metrics to monitor model performance – with the drawback of defining thresholds - was confirmed during the focus group.

> *metrics from the training of the model can be saved. [..] results of the model while it is in production can indicate degrading model performance (INTRESP3).*

> *robust metrics signal that something is going wrong. It does not have to be clear what is going wrong, as long as it signals the need for further investigation. [..] This is difficult [determining what scores are still expected], as to a certain extent you can only make estimates as to what is normal (INTRESP1).*

> *monitor all outputs and from that see if there's any changes that could be a flag, which would then result into more investigation. [..] this is subjective, according to my experience. [..] it's tricky to determine when to alert and when not to alert (FOCRESP2).*

Metrics can be computed on the entire dataset and sub-populations to detect whether there are groups for which it performs poorly. Focus group participants confirmed the effectiveness of metrics over data slices.

> *You can slice it [the model performance] over different parts of your population (INTRESP2).*

> *That can be valuable to do. [..] In my experience doing this has lead to important discoveries regarding data quality topics (FOCRESP2).*

For this performance measuring using metrics, one essential component identified by three interview participants is whether it is possible to retrieve information whether predictions made by an algorithm were correct, as this informs whether direct feedback to assess model performance can be used.

> *The best-case scenario is when you can later see whether what you predicted was accurate. [..] For example, if I forecast the number of purchases next week, after a week I will know how accurate my forecast was. This is direct feedback, but it is not always possible (INTRESP2).*

During the focus group a new method for gaining feedback on whether the model works as expected in cases where access to feedback is unavailable was creating a simulated dataset, for which expected ground-truth is known, and validating new algorithms on this dataset.

> Create *a simulated set of data – with known outcomes - and keep running your model on that one as well. […] it is like a pilot in some sense, sometimes you put them in a simulator to see how they react in different situations (FOCRESP4).*

To summarise, to reach the goal that an organisation should have checks and balances in place that monitor whether an algorithm still works as expected the most essential component to implement is performance monitoring. Direct or indirect feedback can be used to evaluate models over time. These metrics can be sliced over different parts of the dataset to confirm whether models are performing equally well for all subpopulations. The thresholds for model performance can be set and if performance drops below them, an alert is triggered. Determining when to alert is a difficult trade-off, is subjective to a certain extent and differs per use-case. A final viable measure when feedback is unavailable is having a simulated dataset against which the algorithm is continually tested to produce logical results.

4.1.2. Unintended effects

Regarding the goal of **monitoring and preventing unintended effects,** all interview respondents agreed that models would produce unintended effects and predictions when relationships between variables change over time or incorrect input data is given. During interviews, three methods for solving this were given. All interview respondents mentioned controlling of data used for predictions using sanity checks, three interview respondents mentioned controlling the changing of data relationships between variables by examining changes in distributions of data and three mentioned controlling whether input data for a prediction is an outlier compared to data the model was trained on. During the focus group, these methods were confirmed – with the remark that, once again, determining what is a change and what is not is difficult and subjective.

Rule-based checks:

> *You know what your training data looked like and using this knowledge, you can create rules that say all future data must also look like this (INTRESP3).*
>
> *the length of an adult person must be between eighty centimetres and two meters twenty and then always validate new data for being in these boundaries (INTRESP2).*

Controlling data distributions:

> *you have to check if data is changing. [..] any change in the proportions of your input data will change the outcomes of your model (FOCRESP3).*
>
> *controlling data drift you should check that distributions are similar (INTRESP3).*

Controlling outliers:

> *Data that can come in can be a complete outlier compared to what your model was trained on and most likely, your model won't respond well to this. In this case, I would assume that you can get results that are completely out of scope of your expectations also (INTRESP1).*

> *Next to your classifier [model] an out of distribution detector should be added. This detector indicates whether data you are showing it is from the same distribution as the training data. When you have a low probability in this case, that is a sign you do not want to make any predictions for this data at all (INTRESP4).*

During interviews, one respondent noted that to avoid unintended effects, the population distribution should match the distribution in your data. The importance of controlling whether all groups by which the algorithm will be used are represented in training data was confirmed during the focus group.

> *if the entire world exists of 75 percent group A and 25 percent group B, is this also the case in your dataset? (INTRESP3).*

> *you should have balanced input data [..] If your input dataset only has men in it, then it might be biased towards men in decision-making (FOCRESP5).*

One method for preventing unintended effects not mentioned during interviews, but did emerge in the focus group, was controlling whether the data utilised is relevant for predicting the output. For this subject-matter experts can be consulted advice whether variables being used for the algorithm actually make sense conceptually:

> *we once had a dairy trading model for which the price of zinc on world markets was important [..] The correlation is no longer there, but for a long time, milk powder price could be accurately predicted using the price of zinc. [..] An expert in the field can consult whether data you are using to solve the problem makes sense to them. Perhaps they would indicate that you should not use the zinc price if you want to say something about dairy markets (FOCRESP4).*

Overall, the respondents during the interviews and focus group agree that, while preventing all unintended effects is likely impossible, several approaches are valid to prevent unnecessary ones. Rule-based checks should validate whether data is appropriate to make predictions on. Controlling for changing relationships, which triggers an update of the model or whether a data point is out of the distribution of the dataset and, thus, should not receive a prediction are other methods utilised for preventing unintended effects. In this, a difficulty is determining when a change is big enough to trigger actions - this is subjective to a certain extent. Another item worth analysing during development of the model is whether the groups for which the model will be used are represented in the training data. A final method identified that can be utilised during the models development is a reflection with a subject-matter expert on whether the input data is appropriate for predicting the desired output – thus reducing chances of the model utilising spurious relationships which later will turn out to be false.

4.1.3. Bias

The goal that **algorithms cannot contain any bias** was one for which all interview respondents agreed that bias in your input dataset is to be expected. Given this fact, they agree that controlling for bias should happen while constructing the model. All focus group participants agreed with the need to control for unwanted biases during the model development phase – with the remark added that in certain cases bias in outcomes can be desirable, indicating that it depends per use-case which biases should be prevented.

You should do this [checking for bias] during model development. [..] I would think that something went wrong in the process if you find out about biases at a later stage (INTRESP2).

You need to be careful what the end goal is and what consequences of any biases can be. [..] in some cases a bias in a certain direction is desired, as treatments in healthcare for example require differentiation. [..] a lot of it comes down to carefully drafting or scoping your requirements at the start and making sure your model has access to the right data (FOCRESP2).

Three of the interview respondents noted that biases can be controlled through the measuring of fairness metrics to see if the model is treating groups equally. One remarked that in certain cases this is impossible, as GDPR prevents use of the data needed to measure the metrics. During the focus group, the measuring of fairness metrics was confirmed, with the note that which group this has to be measured for must be determined beforehand.

Imagine that you are predicting something expecting the same percentage of positive predictions for both men and women; you can monitor this (INTRESP2).

The detection of unfairness or bias is not difficult in principle. The monitoring techniques available work well for this. [..] A challenge is that according to GDPR in many cases bias is unmeasurable. If you want to know whether you are discriminating minorities, you need data regarding who belongs to which minority. [..] you are not allowed to have this data and if you happen to have it by accident, you are not allowed to touch it (INTRESP3).

An interview respondent noted that a validation dataset can be created, which then sees if an algorithm is biased. This was confirmed during the focus group, where a new method, the utilisation of simple algorithms to approximate outcomes of a black-box model to assesss whether it is biased to sensitive attributes was mentioned also.

A possible test that can be done to detect it [bias] is to create a dataset where you can control explicitly whether predictions have bias or not (INTRESP4).

if you train simple algorithms to approximate outcomes of your complex algorithm and then you can see if there's a bias in your algorithm by looking at these approximations (FOCRESP4).

To summarise, methods to exclude biases from algorithms are the controlling of biases during model development, creating a dataset to explicitly measure potential biases, measuring fairness metrics and utilizing simple models to approximate model outcomes. GDPR seemingly poses an obstacle in certain instances, as model developers cannot use data needed to monitor whether their model has biases.

4.2. Overall system/ model insights

All interview respondents agreed that to reach the goal that **the systems utilising an algorithm must be auditable,** lineage and provenance tracking should be in place. Lineage and provenance tracking involves saving information regarding model settings, automatic and manual changes to models, the input data on which the model was trained, the predictions that are made, the data upon which these predictions were based, the code that trained the model and configured the data to allow insight into the construction of a model and its outputs. This can be used to audit a system or explain unexpected outcomes. An example is well suited to explain the concept:

The concept lineage is important to trace every result - a model, prediction or experiment – to and see how things have occurred. You save the entire heritage of a result. [..] an example in finance. A model is used to predict the future value of a stock and these predictions are used for algorithmic trading. There is a stock for which the model says it will increase with 15 percent the next day. With this information, the algorithmic trading system decides to buy 30 million euros of stock, but the stock actually goes down. In this situation, you want to be able to track down what actually has happened. Of course, you could have been unlucky, but someone can potentially have earned a lot of money by biasing your model. To analyse this, you look at: which model made this prediction and what settings were used for it? On what data was it trained and at what time? Who made it? What code was used to create the data? Using this available information, you can track down all the necessary information and decide that in this case the prediction was bad luck or not (INTRESP2).

The total tracking and versioning of data, models, environment and codes. That should all be tracked together (INTRESP3).

During the focus group, all participants agreed on the usefulness of the lineage and provenance tracking proposed during the interviews for the auditability of algorithms:

if you have all this stuff available on the lineage part and can supply this information for a decision it is probably more elaborate than the average review of a human decision (FOCRESP4).

I think if you log those things it is good enough to audit back to a point of time (FOCRESP5).

One interviews respondent suggested saving versions of training data. During the focus group concerns regarding the feasibility with regards to GDPR restrictions was raised:

if you look at GDPR compliance and consider a person can withdraw any time and you'll need to go through all these datasets to delete the data automatically. I wonder if it's possible (FOCRESP2).

One of the respondents noted that while lineage is a good start, when data is not versioned or saved completely it is not full reproducibility:

To reproduce an individual prediction you have to trace back what happened to it. [..] The exact values for every field in the data need to be identical. This is not easy – partly due to the GDPR retention period (INTRESP3).

To summarise, all participants seemingly agree that lineage and provenance tracking can aid in the making systems using algorithms auditable and providing insight into the full algorithmic lifecycle. Saving versions of training data to be able to reproduce models was met with scepticism during the focus group, as the technology needed for this while adhering to GDPR is complex. A drawback of the suggestions mentioned by respondents during both the interviews and focus group is that lineage and provenance tracking is not identical to reproduciblity – knowing how something came to be is not equal to being able to reproduce it.

4.3. Outcomes

Regarding the goal of having **algorithms that are interpretable,** two main methods came forward during the interview. The first is using models that are explainable by nature. The second method is

the usage of models that are uninterpretable by nature, but whose predictions can be explained using different algorithms. During the interviews, two respondents noted that the quality of these explanations varies greatly, causing a need for quality analysis.

> *Some models are explainable by nature. They are just created that way. So for example for some models there is just a linear contribution of each feature (INTRESP4).*

> *There are many tools to create explanations these days for black-box models. Perhaps you are familiar with the saying "All models are wrong, but some are useful". This is doubly true for explanations, as you start with a model, and all models are wrong, and to that you add another model [the explanation]. All explanations are wrong, but some are useful. [..] You should do quality analysis of explanations (INTRESP3)*

These methods were confirmed during the focus group, but with the side-note that there seems to be a trade-off in projects between using inherently interpretable models – which generally have lower predictive accuracy - or black-box models which are harder to interpret, but have better predictive accuracy. The focus group participants agreed that a use-case determines what kind of interpretability mechanisms are appropriate.

> *The thing that is the real black-box is the world – the world is difficult to understand - and we are building these complex algorithms to match our empirical data. Interpretability is a nice goal, but there is a tradeoff with accuracy in this regard (FOCRESP4).*

> *You can use more generic or more easily explainable model for the sake of it being explainable [..] we've had good feedback in the past with applying generic models knowing that if we couldn't explain what happened, the whole thing would be immediately shut down (FOCRESP2).*

Another aspect of interpretability is giving more than just a prediction as output:

> *You can also include the certainty your model has regarding a prediction. This will show how confident the model is in this prediction (INTRESP4).*

> *You might also want to visualise predictions, as that might highlight remarks (INTRESP1).*

To recapitulate, the main thing that can be done to create interpretable algorithms is the use of inherently interpretable algorithms. Respondents note that applying these methods can have lower predictive performance, but still be preferable over other methods. When more complex models are used, methods to generate explanations for predictions exists, but the quality of these explanations is uncertain, leading to a need for quality analysis. Moreover, methods such as visualisation or giving multiple model outputs can increase the interpretability of the algorithm.

4.4. Human-in-the-loop

Regarding the goal of keeping a **human-in-the-loop** for providing an accountable entity and the monitoring of algorithms, a first possible consideration is regarding the degree of human-control that must be present for a single prediction. Two interview respondents made this distinction, which was confirmed during the focus group with the note that controlling every prediction gets difficult when the number of decisions taken grows rapidly.

You can control predictions after they are used, or manually review every prediction before it is used (INTRESP3)

it's quite difficult to scale up if you have thousands of decisions every hour (FOCRESP5).

Additionally, a human can have control through the decision of whether a model should be deployed. One interview respondent noted that this decision, when taken by senior staff in a company, can perhaps be seen as a digital signature. The participants of the focus group agreed with this statement.

in software engineering you have a production environment and systems that automatically put new versions live. What you don't want is that everyone can just make changes and that these go live, so often this is protected. Someone must send a request to have his changes incorporated into the production environment and there are, for example, two senior people that need to approve these changes before they are incorporated. Something similar can be done for machine learning where any new model, or changes in the code, need to be reviewed by senior staff who get information to make a decision before they are put to production. [..] this signoff can perhaps be seen as a digital signature (INTRESP2).

it [the digital signature] shows people that we have to be responsible. We will put algorithms out there that are making decisions about things that have an impact on people (FOCRESP4).

Another manner of control, confirmed during the focus group, is reactive control, where humans intervene after an alert is triggered:

It can also be more reactive when you have a monitoring in place alerting you if something occurred during the deployment of a new model (INTRESP2).

If it is something new, you need to have some kind of human input or some extra eyes to verify whether it is an anomaly (FOCRESP5).

To summarise, there are different ways of keeping humans in the loop during an algorithm's lifetime, starting with a human reviewing every prediction before it is utilised. This is only doable for modest amounts of predictions, after which a human can be kept in the loop by reviewing predictions reactively or reviewing model deployments. To provide an accountable entity, a workflow with a digital signature before any new algorithm is published can be used.

4.5. Evaluation of the artefact

To assess the **usefulness** and **efficacy** of the artefact the interview respondents were asked for their thoughts on checklist creations that aided in the incorporation of algorithmic accountability through MLM components. During the focus group, a similar question was posed to assess the artefact usefulness and efficacy. All respondents, during both the interviews and focus group, were positive regarding the artefact and thought it would aid professionals in incorporating algorithmic accountability goals into their decision-support systems.

If a checklist like this existed it would definitely be used [..] sometimes in a project the only thing you have in your mind is to build something and ensure it is available [..] other type of questions also need to be answered, so such a document can come in handy in that regard (INTRESP4).

I would use it. It would be useful as you notice there are so many components (INTRESP1).

As an organization, part of the support you want to give teams will indeed be a checklist or best practices (INTRESP3).

it would help hugely [..] perhaps it would not succeed in getting all algorithms in a company accountable within a reasonable period of time [..] we have security policies in place and people still use 'password' as their password. So, I'm unsure if it will succeed completely, but it will definitely aid and educate (FOCRESP4).

It's a good start in getting algorithmic accountability a proper position in the way of working (FOCRESP2).

Several caveats and suggestions to make it more useful were made. One of them being that the degree of algorithmic accountability needed differs per project:

More awareness is important, but some things are not necessary for every project. Some projects are just done to improve processes internally within an organisation. For these projects you would not want a year spent on building these items. The potential positive or negative impact plays an important role in this [..] perhaps a list that is used every project to see which items are needed (INTRESP2).

Two interview respondents mentioned that the checklist could be incorporated into a software tooling at a later point:

it is important that an ecosystem of tools is created that reduce the overhead of incorporating these aspects (INTRESP3).

It would be great if it was available in the form of a software framework (INTRESP4).

Another respondent suggested to not make it too in-depth and refer to another source for more information:

It should be simple language. [..] You should see what to implement for your model and then refer to other sources for more details (INTRESP1).

4.6. Embedding in organisation

The question regarding how much such an **artefact can be best embedded in organisations** was discussed during the focus group. A proposal that was favoured by several participants was seeing the output similar to architectural or security principles in IT, where a third-party in the company reviews whether projects adhere to them.

it should be embedded alike to IT architectural and security principle. Companies need a model accountability team that establishes standards to which you comply if you run a project. We currently only do architectural and security principles because we have them written down and projects are checked against them. Managers do it, even though it's a lot of work, as they have to for projects to go (FOCRESP4).

That's a good idea to make it like a architecture principle (FOCRESP5).

The check should be by a third party within the company, so not by the model builders themselves. And it could be done at the start of a project to see to what extent the checklist must be adhered to and at a later point to see if everything is going as it should (FOCRESP2).

One respondent disagreed with the claim that a third-party should do the control:

Guidelines shoud be set and the teams themselves are responsible for following them. Maybe a team leader is responsible to make sure that the team adheres to them, to reduce dependency on some other team. (FOCRESP5)

5. Discussion, conclusions and recommendations

In this chapter, the results will be discussed and the research questions answered. The main findings will be summarised in the form of propositions. The limitations of the study and outlines for the use of the results in research and practice will also be presented.

5.1. Discussion – reflection and conclusions

This section first discusses the sub-questions, after which the answer to the main research question will be given.

1) SQ1: *Which machine learning monitoring components fit current algorithmic accountability goals?*

During the literature analysis, six accountability goals were determined. For the realisation of each of these suitable MLM components were identified. First, each goal will be discussed below, followed by a general reflection on the findings regarding this sub-question.

- For reaching the goal of **measuring whether an algorithm is performing accurately and as expected** several MLM components from literature that can be used are the following: metrics on an entire dataset, metrics on data slices and alerts notifying an admin. Furthermore, the newly identified components that can be utilised are periodical evaluation by a subject-matter expert and a simulated dataset for which ground-truth answers are known can be utilised.

Nearly everyone during the focus group and interviews agreed on the need to monitor using metrics on direct or indirect feedback to validate algorithms performance. Metrics on the entire dataset and on data slices were identified as appropriate for this – as suggested in literature by (Baylor et al., 2017; Kahng et al., 2016). Respondents agreed that alerts that notify further investigation - as proposed by (Baylor et al., 2017; Polyzotis et al., 2018) – are suitable for keeping control regarding an algorithms performance. An interesting note that emerged during the interviews and was confirmed during the focus group was the need to determine thresholds for alerts. The respondents mention that this is subjective to a certain extent and should be determined per use-case. Another distinction, that developed during the interviews, is the possibility of not having any direct or indirect feedback to validate predictions made by the algorithm. For this, the respondents proposed a new component, the creation of a simulated dataset for which ground-truth answers are known and which the model can continually be evaluated with. A search in literature shows discussion regarding the use of synthetic data to train algorithms in health-care, as in (Chen, Lu, Chen, Williamson, & Mahmood, 2021), but no general framework for creating these datasets and using them to monitor algorithms continually seems to exist as of yet.

- For reaching the goal of **monitoring and prevention of unintended effects,** the MLM components from literature that can be used are sanity checks on the input data and controlling for concept drift. Simultaneously, newly identified components are the controlling of the population distribution, the use of an out-of-distribution detector and the evaluation of the data used by a subject-matter expert.

While respondents seemed to agree that, as in human decision-making, unintended effects and mistakes are not entirely preventable, some measures were identified that may reduce the amount of them in algorithmic decision-making. The most significant distinction found during the interviews and confirmed in the focus group was that unintended effects could be produced when the relations the model was trained on are no longer valid or when the data for which predictions are produced are not comparable to the data on which the algorithm was trained. The most common method for

preventing this is the establishment of sanity checks of the input data, as discussed in (Baylor et al., 2017; Miao et al., 2017; Polyzotis et al., 2018; Sridhar et al., 2018), and the controlling for changes in the data distributions to signal changing relationships between the variables (concept drift), as discussed in (Baylor et al., 2017; Crankshaw et al., 2015; Polyzotis et al., 2018; Sculley et al., 2015). A distinction highlighted during the research is that determining when a change can be considered as such is subjective and needs to be determined per use-case. Another method that was not identified during the literature research, but was proposed during the interviews and confirmed during the focus group, is the creation of an out-of-distribution detector with the goal to determine see whether a data point for which a prediction is made is coming from the same distribution as the training data – and preventing any predictions when this is not the case. A search in literature shows that some research exists for algorithms that have images as input data, as in (Liang, Li, & Srikant, 2018), but for other input data types there are no results. Two final new methods proposed to prevent unintended effects is evaluating whether all eventual users of the algorithm are represented in the training dataset – the distribution in the training data should be representative of the population distribution – and having a subject-matter expert evaluate whether the data being used to predict a certain variable makes sense.

- For reaching the goal that **algorithms cannot contain any bias,** the MLM component from literature that can be used is the use of fairness metrics. Besides, the newly identified useful components are the control for bias during model development, use of a test dataset to flag bias and the use of simple algorithms to approximate complex algorithms outcomes.

For this goal the most important method respondents mentioned – that was suprisingly not found during the literature analysis – was that control for bias should happen during development. All respondents agreed that bias is to be expected and that during the development of an algorithm one should reflect on the ways in which it might be biased and how to monitor this. For this goal one method from the literature was confirmed during both the interviews and focus group, the measuring bias using fairness metrics, as discussed in (Arrieta et al., 2020). Two other methods not present during the literature analysis for identifying bias were proposed: the creation of a test dataset that can flag potential bias in an algorithms results and the utilisation of simple algorithms to explain the outcomes of a complex algorithm to detect whether it is using sensitive attributes influence its predictions. Furthermore, respondents mentioned that GDPR prevents monitoring bias in certain cases, as to measure whether an algorithm is discriminating according to sensitive attributes they need access to these attributes.

- For reaching the goal that **a system using algorithms need to be auditable,** the MLM component from literature that can be used is lineage and provenance tracking

The concept of lineage and provenance tracking - as described in (Miao et al., 2017; Polyzotis et al., 2018; Schelter et al., 2017; Sridhar et al., 2018; Vartak et al., 2016) – was confirmed during both the interviews and the focus group. Nearly all respondents mentioned that with the information provided by saving information regarding algorithm training, historical performance metrics, predictions, algorithm setting, input data, automatic changes, human changes and the code that created the model and transformed the data a system would be auditable for compliance. Respondents during the interviews and focus group agreed that lineage does not provide as much information as full reproducibility – as knowing how something came to be is not identical to being able to reproduce it – but also mention that it is currently often technically infeasible to reproduce results for an indefinite period, as everything in all systems must be in the exact state it was before. Another factor in this is that GDPR in some cases will not allow all data to be stored indefinitely, thus blocking the possibility of reproducibility in certain cases.

- In reaching the goal that **the algorithm must be interpretable,** the MLM components from literature which can be used are the use of inherently explainable models, explainability algorithms and the visualisation of predictions. Furthermore, the newly identified component that can be utilised was the inclusion of model certainty in predictions.

The most common method mentioned by respondents during both the interviews and the focus group was the utilisation of inherently interpretable models. There seems to be a tradeoff in this regard, where black-box methods utilise have more predictive power and inherently interpretable models have the benefit of being interpretable by design instead of needing explainability algorithms. What is needed in this regard is determined per use-case. Respondents mention that when black-box methods are used, exaplainability algortihms can be applied to infer what the algorithm bases its decisions on. They noted that the quality of these explanations varies, causing a need for a quality analysis of predictions. Visualising predictions as a way to interpret a model – as discussed in (Kahng et al., 2016; Polyzotis et al., 2018; Sculley et al., 2015) – was confirmed during both the interviews and focus groups as a suitable method for enhancing interpretability of models. A new method for enhancing the interpretability of a prediction mentioned was to, next to the prediction, include how certain the model was about its prediction.

- For reaching the goal of keeping a **human-in-the-loop,** the MLM components from literature that can be utilised include the controlling of predictions before or after they are used, the controlling of model deployments or the reactive controlling of predictions. Additionally, the newly identified component that can be employed is the creation of a model deployment flow where a digital signature is given to deploy a model.

A balance that emerged for this goal was between the quantity of predictions and the effort that can be taken in reviewing them. When the number of predictions is low – such as seen in some healthcare cases - an expert can review every prediction before its use, as discussed in (Baylor et al., 2017). When the number of predictions becomes too high, predictions can only be reviewed reactively, as discussed in (Baylor et al., 2017), or the decision changes from each prediction to the deployment of models, as discussed in (Sridhar et al., 2018). A new method providing a manner of having an accountable entity with regards to algorithm deployment, that was found during the interviews and confirmed during the focus group, was having a signoff where a senior staff member receives information regarding an algorithm – or potential changes made to an algorithm – and must approve it before these changes are published. This signoff can be seen as a digital signature. This ensures a certain responsibility attached to the act of deploying an algorithm which makes decisions and influences society. A search in literature shows that the use of the suggested system (Git) has not been researched for this purpose yet. Similar research can perhaps be Prieto, Izkara, and Béjar (2018), where the system is used to maintain accurate 3D models of cities.

In conclusion, for each algorithmic accountability goal several MLM components were found. Noteworthy is that for nearly all goals new components that were not identified in the literature search were found – namely, simulating data for measuring model performance, out-of-distribution detectors to prevent unintended effects, the importance of steps in the process to prevent biases and unintended effects, test datasets to detect biases, including model certainty to enhance interpretability and the creation of a system with a digital signature for model signoff to provide an accountable entity. Moreover, while the frameworks analysed for components in the literature research all describe only technical components, in certain cases the respondents found the components relating to the process the most important. This can indicate that future MLM frameworks should incorporate these type of components more explicitly if they intent on being

suitable for achieving algorithmic accountability. Another finding is that, in two cases, monitoring bias and reproducibility, GDPR seeminlgy prevents professionals from achieving algorithmic accountability.

2) SQ2: *How could an artefact incorporating these components be embedded in an organisation so as to achieve algorithmic accountability in practice?*

The suitability of the proposed checklist, which can be found in Appendix 5, was evaluated during both the interviews and the focus group. During the interviews, the focus was on what requirements the checklist should have, while during the focus group emphasis was on how to embed such a checklist in an organisation, whether any alternatives existed and the evaluation of the artefact on the chosen metrics efficacy and usefulness.

- The proposed artefact can best be embedded in organisations similarly to current architectural or security principles.

During the focus group, all respondents agreed that the embedding of the artefact in organisations can most effectively be orchestrated in a similar strain to current architectural or security principles. In this regard, organisations could have a person, or team, responsible for formulating algorithmic accountability guidelines and methods - using an artefact such as the proposed checklist – to which AI projects in the company need to adhere. This will mean that a project is discontinued if it does not adhere to the set guidelines. There was a slight disagreement in the focus group regarding whether a third party within the company, or the teams themselves, should evaluate their projects using the checklist. Proponents of a third party within the company argue that the model builders themselves should not evaluate their work, while detractors argue that such a third party is likely to run into long waiting times before a project is approved. Both the interviews and the focus group respondents mentioned that per project the necessary MLM components should be determined, to prevent unnecessary effort during implementation. Two interview respondents noted that the checklist could perhaps at a later point be complemented by software, which can reduce the effort necessary to implement the methods.

- The proposed artefact can aid professionals in incorporating algorithmic accountability goals within their decision-support systems

The **efficacy** and **usefulness** of the artefact were evaluated during both the interviews and focus group. During the interviews, respondents were asked for their opinion on the proposed checklist. All respondents indicated the potential benefits of using such an artefact in the incorporation of algorithmic accountability in decision-support systems. During the focus group the efficacy and usefulness were evaluated by the question, when embedded in an organisation as discussed during the focus group, how much the respondents thought the artefact would aid professionals to incorporate algorithmic accountability goals into their systems. Overall, the results were positive, with respondents noting that while the introduction of such a checklist might not make all algorithms accountable immediately, it would certainly help and educate.

Concluding, when evaluated on the metrics efficacy and usefulness, the proposed checklist – when embedded in a similar way as architectural and security principles – seems to be suitable in aiding professionals in incorporating algorithmic accountability goals into their decision-support systems. Whether it should be within the teams themselves, or a third-party within the organisations that reflect on their projects using the guidelines that can be set with the checklist, appears open for debate. Whether the format of a checklist is the best for the artefact is an item that seems to be open

for discussion, some participants liked the concept, while others suggested that a format like software is better suited.

RQ: *"How can a machine learning monitoring methodology be incorporated into decision-support systems so as to achieve algorithmic accountability?"*

A solution to the main research question lies in the answers to the previous sub-questions. A MLM methodology can be incorporated into decision-support systems so as to achieve algorithmic accountability by the use of the methods included in the created artefact – given in Appendix 5. An organisation can achieve this by basing the embedding in the organisation in a similar way to architectural and security principles, where guidelines and methods to be used are set by the company and projects evaluated based on them. It is unclear to what extent algorithmic accountability will be achieved through the implementation of the artefact in this manner. However, the results indicate that the incorporation of MLM methods can aid in the realisation of algorithmic accountability goals, as was hypothesised at the start of the research.

5.2. Limitations

Every research approach has its limitations. This section discusses the ones encountered in this approach.

An initial limitation is that the approach did not research whether algorithmic accountability may in actual fact be achieved by incorporating the mentioned components. Due to the exploratory nature of the research, where the methods for achieving algorithmic accountability goals still had to be discovered and confirmed, the direct effect of them on reaching these goals could not be measured. Alternatively, the knowledge of professionals was taken as an indication that the methods would be effective in this regard.

A further limitation was that the research had a limited sample of nine respondents. In the future, the sample can be enlarged or other methods used to triangulate the results. Due to time constraints, more iterative rounds of data collection and analysis were also not possible. Additionally, all participants were professionals in the Netherlands, which might mean that there is a bias as to methods that are more common in this context.

Finally, quite a narrow definition of algorithmic accountability was used in this research. The focus was on how an organisation that intends to keep its algorithms accountable can reach this through the use of MLM methodology. A side-effect of this is that it ignores part of the algorithmic accountability field, which, for example, focuses on how organisations not wanting to keep their algorithms accountable should be handled.

5.3. Academic relevance

The main academic relevance seems to be in the fact that MLM components can be used in the realisation of algorithmic accountability goals. Several of the methods found - simulating data for measuring model performance, out-of-distribution detectors to prevent unintended effects, the importance of steps in the process to prevent biases, test datasets to detect biases, including model certainty to enhance interpretability and the creation of a system with a digital signature for model signoff to provide an accountable entity - were not identified during the literature research regarding MLM either, so can be regarded as new for methods for the field of algorithmic accountability. Combining the new and confirmed methods, the proposed artefact contains a list of methods that can be used to realise the found algorithmic accountability goals. Other relevant findings entail the viewing

of algorithmic accountability principles in a similar strain to security principles, a more mature field to which machine learning is perhaps more similar than the comparison to healthcare in B. Mittelstadt (2019), as this is also a field into where certain methods and processes need to be implemented to use technology safely. Learnings taken from this field regarding how to implement principles into organisations can be transferred to the field of algorithmic accountability. A final finding of interest for the field of algorithmic accountability is that several respondents mentioned two instances in which GDPR blocks them from keeping their algorithms accountable – in the cases of monitoring for bias and reproducing earlier results.

5.4. Recommendations for practice and further research

Based on the results, several recommendations can be made for practice and further research.

Recommendations for practice:

- By embedding the artefact in Appendix 5 in a similar manner as security principles, where organisations have a person, or a team, using the artefact to establish guidelines and methods that should be used, organisations can start incorporating algorithmic accountability into their decision-support systems.

Recommendations for further research:

- This research has produced a checklist in the form of algorithmic accountability goals and methods contributing to realising them. Future iterations of the design science research cycle can try to further validate this list by testing how these methods impact the realisation of these goals, or algorithmic accountability, directly.
- Due to the research approach taken, the implementation of the artefact in an organisation was not tested directly yet. Future iterations of the design science research cycle can take the form of researching how this can be done optimally. Indicative directions are the suggested approach based on architectural and security principles and the form of software as suggested by two participants.
- Further research could be done to investigate the claims that emerged that, in certain cases, the GDPR blocks organisations from successfully keeping their algorithms accountable, with focus on the mentioned topics of monitoring bias and reproducing results.

References

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., . . . Benjamins, R. (2020). Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion, 58*, 82-115.

Baskerville, R., Baiyere, A., Gregor, S., Hevner, A., & Rossi, M. (2018). Design science research contributions: finding a balance between artifact and theory. *Journal of the Association for Information Systems, 19*(5), 3.

Baylor, D., Breck, E., Cheng, H.-T., Fiedel, N., Foo, C. Y., Haque, Z., . . . Koc, L. (2017). *Tfx: A tensorflow-based production-scale machine learning platform.* Paper presented at the Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Binns, R. (2018). Algorithmic accountability and public reason. *Philosophy & Technology, 31*(4), 543-556.

Chen, R. J., Lu, M. Y., Chen, T. Y., Williamson, D. F., & Mahmood, F. (2021). Synthetic data in machine learning for medicine and healthcare. *Nature Biomedical Engineering*, 1-5.

Crankshaw, D., Bailis, P., Gonzalez, J. E., Li, H., Zhang, Z., Franklin, M. J., . . . Jordan, M. I. (2015). *The Missing Piece in Complex Analytics: Low Latency, Scalable Model Management and Serving with Velox.* Paper presented at the Proceedings of the Biennial Conference on Innovative Data Systems Research.

Danaher, J., Hogan, M. J., Noone, C., Kennedy, R., Behan, A., De Paor, A., . . . Morison, J. (2017). Algorithmic governance: Developing a research agenda through the power of collective intelligence. *Big Data & Society, 4*(2), 2053951717726554.

De Laat, P. B. (2018). Algorithmic decision-making based on machine learning from Big Data: Can transparency restore accountability? *Philosophy & Technology, 31*(4), 525-541.

Greene, D., Hoffmann, A. L., & Stark, L. (2019). *Better, nicer, clearer, fairer: A critical assessment of the movement for ethical artificial intelligence and machine learning.* Paper presented at the Proceedings of the 52nd Hawaii International Conference on System Sciences.

Hevner, A. R. (2007). A three cycle view of design science research. *Scandinavian journal of information systems, 19*(2), 4.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS quarterly*, 75-105.

Kahng, M., Fang, D., & Chau, D. H. (2016). *Visual exploration of machine learning results using data cube analysis.* Paper presented at the Proceedings of the Workshop on Human-In-the-Loop Data Analytics.

Kitzinger, J. (1994). The methodology of focus groups: the importance of interaction between research participants. *Sociology of health & illness, 16*(1), 103-121.

Lepri, B., Oliver, N., Letouzé, E., Pentland, A., & Vinck, P. (2018). Fair, transparent, and accountable algorithmic decision-making processes. *Philosophy & Technology, 31*(4), 611-627.

Liang, S., Li, Y., & Srikant, R. (2018). *Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks.* Paper presented at the International Conference on Learning Representations.

Lincoln, Y. S., & Guba, E. G. (1986). But is it rigorous? Trustworthiness and authenticity in naturalistic evaluation. *New directions for program evaluation, 1986*(30), 73-84.

McGregor, L., Murray, D., & Ng, V. (2019). International human rights law as a framework for algorithmic accountability. *International & Comparative Law Quarterly, 68*(2), 309-343.

Miao, H., Li, A., Davis, L. S., & Deshpande, A. (2017). *Towards unified data and lifecycle management for deep learning.* Paper presented at the 2017 IEEE 33rd International Conference on Data Engineering (ICDE).

Mijač, M. (2019). *Evaluation of Design Science instantiation artifacts in Software engineering research.* Paper presented at the Central European Conference on Information and Intelligent Systems.

Mittelstadt, B. (2019). Principles alone cannot guarantee ethical AI. *Nature Machine Intelligence, 1*(11), 501-507.

Mittelstadt, B. D., Allo, P., Taddeo, M., Wachter, S., & Floridi, L. (2016). The ethics of algorithms: Mapping the debate. *Big Data & Society, 3*(2), 2053951716679679.

Morgan, D. L. (1996). FOCUS GROUPS. *Annu. Rev. Sociol, 22*, 129-152.

Offermann, P., Blom, S., Schönherr, M., & Bub, U. (2010). *Artifact types in information systems design science–a literature review.* Paper presented at the International Conference on Design Science Research in Information Systems.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems, 24*(3), 45-77.

Polyzotis, N., Roy, S., Whang, S. E., & Zinkevich, M. (2018). Data lifecycle challenges in production machine learning: a survey. *ACM SIGMOD Record, 47*(2), 17-28.

Prieto, I., Izkara, J. L., & Béjar, R. (2018). A continuous deployment-based approach for the collaborative creation, maintenance, testing and deployment of CityGML models. *International Journal of Geographical Information Science, 32*(2), 282-301.

Rahwan, I. (2018). Society-in-the-loop: programming the algorithmic social contract. *Ethics and Information Technology, 20*(1), 5-14.

Raji, I. D., Smart, A., White, R. N., Mitchell, M., Gebru, T., Hutchinson, B., . . . Barnes, P. (2020). *Closing the AI accountability gap: defining an end-to-end framework for internal algorithmic auditing.* Paper presented at the Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency.

Schelter, S., Boese, J.-H., Kirschnick, J., Klein, T., & Seufert, S. (2017). *Automatically tracking metadata and provenance of machine learning experiments.* Paper presented at the Advances in neural information processing systems.

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., . . . Dennison, D. (2015). *Hidden technical debt in machine learning systems.* Paper presented at the Advances in neural information processing systems.

Shin, D., & Park, Y. J. (2019). Role of fairness, accountability, and transparency in algorithmic affordance. *Computers in Human Behavior, 98*, 277-284.

Sridhar, V., Subramanian, S., Arteaga, D., Sundararaman, S., Roselli, D., & Talagala, N. (2018). *Model governance: Reducing the anarchy of production {ML}.* Paper presented at the 2018 {USENIX} Annual Technical Conference ({USENIX}{ATC} 18).

Thornhill, A., Saunders, M., & Lewis, P. (2009). Research methods for business students. *Essex: Pearson Education Ltd*.

Vartak, M., Subramanyam, H., Lee, W.-E., Viswanathan, S., Husnoo, S., Madden, S., & Zaharia, M. (2016). *MODELDB: A System for Machine Learning Model Management.* Paper presented at the Proceedings of the Workshop on Human-In-the-Loop Data Analytics.

Wolfswinkel, J. F. (2011). Using grounded theory as a method for rigorously reviewing literature. *European Journal of Information Systems, 1*, 11.

Zarsky, T. (2016). The trouble with algorithmic decisions: An analytic road map to examine efficiency and fairness in automated and opaque decision making. *Science, Technology, & Human Values, 41*(1), 118-132.

Appendix 1: (Sub)category and concept table machine learning monitoring

Table A1.1 below displays for each MLM component found the relevant literature discussing it.

Table A1.1: matching of each category, sub-category and concept and the literature discussing it

Category	Sub-category	Concept	Articles
Assessing the appropriateness of data	Data validation	Controlling for training-serving skew	(Baylor et al., 2017; Crankshaw et al., 2015; Polyzotis et al., 2018)
Assessing the appropriateness of data	Data validation	Sanity checks	(Baylor et al., 2017; Miao et al., 2017; Polyzotis et al., 2018; Sridhar et al., 2018)
Assessing the appropriateness of data	Data validation	Statistics for data change detection	(Baylor et al., 2017; Polyzotis et al., 2018; Sculley et al., 2015)
Assessing the appropriateness of data	Updating models with new data	Automated model training	(Baylor et al., 2017; Crankshaw et al., 2015; Miao et al., 2017; Polyzotis et al., 2018; Sridhar et al., 2018; Vartak et al., 2016)
Model Validation	Human	Alerts that notify admin	(Baylor et al., 2017; Polyzotis et al., 2018)
Model validation	Human	Human in the loop	(Baylor et al., 2017; Sridhar et al., 2018)
Model validation	Metrics	Fairness / bias adjustments in training	(Arrieta et al., 2020)
Model validation	Metrics	Metrics on data slices	(Baylor et al., 2017; Kahng et al., 2016)
Model validation	Metrics	Metrics on entire dataset	(Baylor et al., 2017; Crankshaw et al., 2015; Miao et al., 2017; Polyzotis et al., 2018; Sridhar et al., 2018; Vartak et al., 2016)
Model validation	Predictions	Detect changes in prediction behaviour	(Sculley et al., 2015)
Model validation	Predictions	Visual exploration of predictions	(Kahng et al., 2016; Polyzotis et al., 2018; Sculley et al., 2015)
Reproducibility & auditability	Metadata	Configuration framework	(Baylor et al., 2017; Sridhar et al., 2018)
Reproducibility & auditability	Lineage & Provenance	Lineage storage	(Miao et al., 2017; Sridhar et al., 2018)
Reproducibility & auditability	Metadata	Metadata management	(Polyzotis et al., 2018; Schelter et al., 2017; Sridhar et al., 2018; Vartak et al., 2016)
Reproducibility & auditability	Metadata	Metric tracking and storage	(Schelter et al., 2017; Sculley et al., 2015; Sridhar et al., 2018; Vartak et al., 2016)
Reproducibility & auditability	Reproducibility	Ability to reproduce predictions	(Sridhar et al., 2018)
Reproducibility & auditability	Lineage & Provenance	Provenance	(Polyzotis et al., 2018; Schelter et al., 2017; Sridhar et al., 2018)
Reproducibility & auditability	Reproducibility	Reproducible experiments	(Schelter et al., 2017; Sculley et al., 2015; Sridhar et al., 2018; Vartak et al., 2016)

Appendix 2: Interview protocol

1. Introduction

- Permission to record
- Discussion of research questions, goals and approach.
- Could you introduce yourself?

2. Clarification of terminology

- Discussion of the definitions listed in the table of definitions:

Term	Meaning
Algorithmic accountability	The degree to which it is possible to hold an algorithm accountable for the predictions it has made (Lepri et al., 2018).
Data science / machine learning algorithm	A data science / machine learning algorithm is an algorithm that takes a bottom-up approach and extracts rules that it learns from data and uses it for future predictions (McGregor et al., 2019).
Decision-support system	A decision-support system is a system that utilises machine learning algorithms to make predictions and decisions, with little or no human input (Binns, 2018).
Machine learning monitoring	"The ability to determine the creation path, subsequent usage, and consequent outcomes of an ML model, and the use of this information to accomplish a range of tasks including reproducing and diagnosing problems and enforcing compliance" (Sridhar, Subramanian et al. 2018, p. 351).
Machine learning monitoring methodology / component	The application of one (or more) machine learning monitoring methods (Baylor et al., 2017).

3. Goal of interview

- The goal of the interview is discussed:
    - The goal of this interview is to gather requirements for and evaluate a checklist of machine learning monitoring components that can aid in the incorporation of algorithmic accountability within decision-support systems.
    - We will discuss each algorithmic accountability goal and then the identified machine learning monitoring components that can be incorporated into decision-support systems to achieve this goal.
    - After this we will evaluate the artefact on two metrics, namely:
        - Efficacy: the degree to which the artefact achieves its goal considered narrowly, without addressing situational concerns.
        - Usefulness: the degree to which the artefact positively impacts the task performance of individuals.

4. Protocol per algorithmic accountability goal

- Per goal:
    - Introduction goal
    - **Question**: "Do you have any questions regarding this goal?"
        - If answered yes, continue
        - If answered no, elaborate on the goal
    - **Question**: "What should be implemented, or practices should be done, in order to realise this goal?"

5. **Protocol for evaluation of the artefact**
    - **Question:** “to which extent do you think the artefact realises the goal of enabling the incorporation of algorithmic accountability within decision-support systems?”
    - **Question: “**Which requirements do you think this artefact has to adhere to?**”**
    - **Question: “**What adjustments should be made to improve the artefact?**”**

6. **Closing**
    - **Question:** “Do you have any other remarks or questions?”
    - **Closing**: I would like to thank you for your participation in the research. At a later point I will contact you with the conclusions I have drawn from this interview so you can review them and still have the option to opt out of the research.

Appendix 3: Focus group protocol

1. Introduction

- Permission to record
- Discussion of research questions, goals and approach.
- Could you introduce yourself?

2. Clarification of terminology

- Discussion of the definitions listed in the table of definitions:

Term	Meaning
Algorithmic accountability	The degree to which it is possible to hold an algorithm accountable for the predictions it has made (Lepri et al., 2018).
Data science / machine learning algorithm	A data science / machine learning algorithm is an algorithm that takes a bottom-up approach and extracts rules that it learns from data and uses it for future predictions (McGregor et al., 2019).
Decision-support system	A decision-support system is a system that utilises machine learning algorithms to make predictions and decisions, with little or no human input (Binns, 2018).
Machine learning monitoring	"The ability to determine the creation path, subsequent usage, and consequent outcomes of an ML model, and the use of this information to accomplish a range of tasks including reproducing and diagnosing problems and enforcing compliance" (Sridhar, Subramanian et al. 2018, p. 351).
Machine learning monitoring methodology / component	The application of one (or more) machine learning monitoring methods (Baylor et al., 2017).

3. Goal of focus group

- The goal of the focus group is discussed:
    - The goal of this focus group is to gather evaluate a checklist of machine learning monitoring components that can aid in the incorporation of algorithmic accountability within decision-support systems.
    - We will discuss each algorithmic accountability goal and then the identified machine learning monitoring components that can be incorporated into decision-support systems to achieve this goal.
    - After this we will evaluate the artefact on two metrics, namely:
        - Efficacy: the degree to which the artefact achieves its goal considered narrowly, without addressing situational concerns.
        - Usefulness: the degree to which the artefact positively impacts the task performance of individuals.
    - We will end with a discussion of results found during interviews and space for any remarks or questions.

4. Protocol per algorithmic accountability goal

- Per goal:
    - Introduction goal
    - **Question**: "Does anyone have any questions regarding this goal?"
        - If answered yes, continue

        - If answered no, elaborate on the goal
    - Per machine learning monitoring component that was found during the interviews:
        - Introduction component
        - **Question**: “Does anyone have any questions regarding this component?”
            - If answered yes, continue
            - If answered no, elaborate on the component
        - **Question**: “Why do you believe this component does (not) aid in the achievement of the goal?”
            - Answer per participant
            - Discussion between participants
    - **Question**: “To your knowledge are there any components missing from this list?’
        - Answer per participant
        - Discussion between participants

5. **Protocol for evaluation of the artefact**
    - **Question:** “How should the artefact be embedded in an organisation so as to achieve algorithmic accountability?”
    - **Question:** “To what degree do you think such a research output can aid professionals in incorporating algorithmic accountability goals into their systems?”
        - Answer per participant
        - Discussion between participants
    - **Question: “**What adjustments should be made to improve the artefact?**”**
        - Answer per participant
        - Discussion between participants
7. **Closing**
    - **Question:** “Do you have any other remarks or questions?”
    - **Closing**: I would like to thank you for your participation in the research. At a later point I will contact each of you personally with the conclusions I have drawn from this focus group so you can review them and still have the option to opt out of the research.

Appendix 4: Interview and focus group coding analysis tables

Below are given the two summary tables of results that came forward during the coding of the data. Table A4.1 contains the different components that emerged during the interviews and focus group.

Table A4.1 MLM components that emerged from the interviews and focus group

Group	Sub-group	Respondent interviews	Respondent focus group
Bias	Control for bias should happen during development	INTRESP1, INTRESP2, INTRESP3, INTRESP4	FOCRESP2, FOCRESP3, FOCRESP4, FOCRESP5, FOCRESP6
	Creation of a test dataset to flag potential bias	INTRESP4	FOCRESP3, FOCRESP4
	Measure fairness using metric	INTRESP2, INTRESP3, INTRESP4	FOCRESP4, FOCRESP2, FOCRESP3
	Use-case dependent choice of fairness metric	INTRESP3	
	GDPR check for possibilitty of checking for bias	INTRESP3	
	Check whether population distribution is represented in dataset	INTRESP3	FOCRESP4, FOCRESP5, FOCRESP2
	Determine possible cases of general influence	INTRESP3	
	Simple models to give insight into complex models their outcomes		FOCRESP4
Lineage and provenance tracking	Saving information regarding model	INTRESP1, INTRESP2, INTRESP3, INTRESP4	FOCRESP3, FOCRESP2, FOCRESP4, FOCRESP5
	Saving information regarding historical performance metrics	INTRESP1, INTRESP2, INTRESP3	FOCRESP3, FOCRESP2, FOCRESP4, FOCRESP5
	Saving information regarding predictions	INTRESP1, INTRESP2, INTRESP3, INTRESP4	FOCRESP3, FOCRESP2, FOCRESP4, FOCRESP5
	Saving information regarding model settings	INTRESP1, INTRESP2, INTRESP3, INTRESP4	FOCRESP3, FOCRESP2, FOCRESP4, FOCRESP5
	Saving information regarding input data	INTRESP1, INTRESP2, INTRESP3, INTRESP4	FOCRESP3, FOCRESP2, FOCRESP4, FOCRESP5
	Save information regarding all automatic and human changes made to models	INTRESP2, INTRESP4	FOCRESP3, FOCRESP2, FOCRESP4, FOCRESP5
	Save code that created model and transformed data	INTRESP1, INTRESP2, INTRESP3, INTRESP4	FOCRESP3, FOCRESP2, FOCRESP4, FOCRESP5
	Knowledge needed to interpret and analyse predictions	INTRESP1, INTRESP3	
	Limiting access and logging who has it	INTRESP2	
Performance monitoring	Metrics on dataset	INTRESP1, INTRESP2, INTRESP3, INTRESP4	FOCRESP4, FOCRESP2
	Metrics on data slices	INTRESP1, INTRESP2, INTRESP3	FOCRESP4, FOCRESP2
	Control of the prediction distribution	INTRESP1, INTRESP2, INTRESP3	FOCRESP3, FOCRESP4
	Visual exploration of predictions	INTRESP1, INTRESP4	FOCRESP3
	Alerts that notify an admin	INTRESP1, INTRESP2, INTRESP3	FOCRESP2, FOCRESP4
	Control of model assumptions	INTRESP1, INTRESP2	
	Determination of thresholds	INTRESP1, INTRESP3	FOCRESP2, FOCRESP4
	Simulated data to validate model on		FOCRESP4
Data assessment	Controlling data distributions	INTRESP2, INTRESP3, INTRESP4	FOCRESP2, FOCRESP3, FOCRESP4, FOCRESP4, FOCRESP6
	Rule based sanity checks	INTRESP1, INTRESP2, INTRESP3, INTRESP4	FOCRESP2, FOCRESP3, FOCRESP4, FOCRESP4, FOCRESP6
	Edge case determination	INTRESP4	FOCRESP2, FOCRESP3
	Out of distribution detector	INTRESP4. INTRESP2, INTRESP3	FOCRESP3, FOCRESP4
	Subject-matter expert to validate whether data used makes sense		FOCRESP4
Human in the loop	Control predictions before use	INTRESP2, INTRESP3	FOCRESP4, FOCRESP5, FOCRESP2
	Control predictions after use	INTRESP1, INTRESP2, INTRESP3	
	Control model deployment with a signoff	INTRESP2, INTRESP4	FOCRESP3, FOCRESP2, FOCRESP5, FOCRESP4, FOCRESP6
	Control based on alerts	INTRESP1, INTRESP2	FOCRESP5, FOCRESP2
Interpretability	Utilise inherently interpretable models	INTRESP3, INTRESP4	FOCRESP2, FOCRESP3, FOCRESP4
	Use of explainability algorithms for blackbox models (choice needed)	INTRESP2, INTRESP3	FOCRESP4

	Quality analysis of explanations	INTRESP2, INTRESP3	FOCRESP4
	Including model certainty in predictions	INTRESP4	
	Visualise predictions	INTRESP1, INTRESP4	FOCRESP2
	Including feature influence per predictions	INTRESP2, INTRESP4	
	Knowledge of project needs to be documented to facilitate intrepratability of results	INTRESP1, INTRESP3	

Table A4.2 shows which literature concepts they match.

Table A4.2 matching table of MLM components that emerged from interviews and literature

Group	Sub-group	Matches MLM concept	Literature
Bias detection	Control for bias should happen during development	New	-
	Creation of a test dataset to flag potential bias	New	-
	Measure fairness using metric	Fairness / bias adjustments in training	(Arrieta et al., 2020)
	Use-case dependent choice of fairness metric	New	-
	GDPR check for possibilitty of checking for bias	New	-
	Check whether population distribution is represented in dataset	New	-
	Determine possible cases of general influence	New	-
	Simple models to give insight into complex models their outcomes	New	
Lineage and provenance tracking	Saving information regarding model	Lineage & Provenance, Metadata management, metric tracking and storage, reproducible experiments	(Miao et al., 2017; Polyzotis et al., 2018; Schelter et al., 2017; Sridhar et al., 2018; Vartak et al., 2016)
	Saving information regarding historical performance metrics		
	Saving information regarding predictions		
	Saving information regarding model settings		
	Saving information regarding input data		
	Save information regarding all automatic and human changes made to models		
	Save code that created model and transformed data		
	Knowledge needed to interpret and analyse predictions	New	-
	Limiting access and logging who has it	New	-
Performance monitoring	Metrics on dataset	Metrics on dataset	(Baylor et al., 2017; Crankshaw et al., 2015; Miao et al., 2017; Polyzotis et al., 2018; Sridhar et al., 2018; Vartak et al., 2016)
	Metrics on data slices	Metrics on data slices	(Baylor et al., 2017; Kahng et al., 2016)
	Control of the prediction distribution	Detect changes in prediction behaviour	(Sculley et al., 2015)
	Expert can judge predictions	New	-
	Visual exploration of predictions	Visual exploration of predictions	(Kahng et al., 2016; Polyzotis et al., 2018; Sculley et al., 2015)
	Alerts that notify an admin	Alerts that notify admin	(Baylor et al., 2017; Polyzotis et al., 2018)

	Control of model assumptions	New	-
	Determination of thresholds	Data cube analysis to determine sanity checks	(Polyzotis et al., 2018)
	Simulated data to validate model on	New	
Data assessment	Controlling data distributions	Controlling for training-serving skew, statistics for data change detection	(Baylor et al., 2017; Crankshaw et al., 2015; Polyzotis et al., 2018; Sculley et al., 2015)
	Rule based sanity checks	Sanity checks	(Baylor et al., 2017; Miao et al., 2017; Polyzotis et al., 2018; Sridhar et al., 2018)
	Edge case determination	New	-
	Out of distribution detector	New	-
	Subject-matter expert to validate whether data used makes sense		
Human in the loop	Control predictions before use	Human-in-the-loop	(Baylor et al., 2017; Sridhar et al., 2018)
	Control predictions after use	Human-in-the-loop	(Baylor et al., 2017; Sridhar et al., 2018)
	Confirm model deployment	Human-in-the-loop	(Baylor et al., 2017; Sridhar et al., 2018)
	Control model deployment with a signoff that provides accountability	New	-
	Control based on alerts	Human-in-the-loop, alerts that notify an admin	(Baylor et al., 2017; Sridhar et al., 2018)
Interpretability	Utilise inherently interpretable models	New	-
	Use of explainability algorithms for blackbox models (choice needed)	New	-
	Quality analysis of explanations	New	-
	Including model certainty in predictions	New	-
	Visualise predictions	Visual exploration of predictions	(Kahng et al., 2016; Polyzotis et al., 2018; Sculley et al., 2015)
	Including feature influence per predictions	New	-
	Knowledge of project needs to be documented to facilitate intrepratability of results	New	-

Appendix 5: Proposed artefact / checklist

The table below shows the proposed artefact in the form of a checklist. Organisations can utilise this artefact by determining per project which of the machine learning monitoring components should be implemented.

Algorithmic accountability goal	**Machine learning monitoring component**	**Literature references**	**Comment / remarks/ drawbacks**	**Applicable for project yes/ no**
Monitoring and preventing unintended effects	Controlling data distributions	(Baylor et al., 2017; Crankshaw et al., 2015; Polyzotis et al., 2018; Sculley et al., 2015)	- "The underlying process can really change [concept drift], for example, you're modelling human behaviour and people get accustomed to something or they slowly start doing things in a different manner. So, for controlling drift you should check that the distributions are similar."	
	Rule based sanity checks	(Baylor et al., 2017; Miao et al., 2017; Polyzotis et al., 2018; Sridhar et al., 2018)	- "You know what your training data looked like as you've analysed this and using this knowledge you can create rules. These rules say that all future data must also look like this." - "A classic example is the length of a variable changing from meters to millimetres, this wouldn't cause an error but if your model was trained on meters it will just make nonsense predictions. This can be aided by adding sanity checks. For example, you can say that the length of an adult person must be between eighty centimetres and two meters twenty and then always validate new data for being in these boundaries."	
	Out of distribution detector	-	- "So, what you should add next to your classifier [model] is an out of distribution detector. The only thing this detector will have to do is note whether the data you are showing is from this distribution or not – the distribution meaning the same distribution as the data your model was trained on. When you have a low likelihood in this case, that is a sign you do not want to make any predictions for this data at all"	
	Check whether population distribution is represented in dataset	-	- " So if the entire world exists of 75 percent group A and 25 percent group B, is this also the case in your dataset?" - "you should have a balanced input data set [..] If your input dataset only has men in it, then it might be biased towards men in the decision making." - "it depends also on what do you want to get out of it. […] for example, if medicine is tested only on men, then it's unclear whether the outcomes are representative for women."	
	Have a subject-matter expert validate whether logical data is being used	-	- "So what about expert knowledge? Having an expert in the field during the design phase consult whether the data you are using to solve the problem makes sense to them."	
Measuring whether an algorithm is performing accurately and as expected	Metrics on dataset	(Baylor et al., 2017; Crankshaw et al., 2015; Miao et al., 2017; Polyzotis et al., 2018; Sridhar et al.,	- "The most direct way is to get feedback from the underlying system that is using the predictions so you can compare the actual performance of the algorithm with what is expected" - "All metrics from the training of the model can be saved. After this for all predictions the outputs can be saved so that the results of the model while it is in production can be compared with earlier results and rolling averages and standard deviations can then indicate degrading model performance."	

		2018; Vartak et al., 2016)		
	Metrics on data slices	(Baylor et al., 2017; Kahng et al., 2016)	- You can slice it [the model performance] over different parts of your population" - "You save it [the model performance] and visualize it over different cross-sections"	
	Control of the prediction distribution	(Sculley et al., 2015)	- "You can monitor whether the distribution of the values that are predicted is different in production than during the training of the model"	
	Determination of thresholds	(Polyzotis et al., 2018)	- "You have a monitoring system in place which is not perfect. So, a solution for this is that it sends an alert that notifies a potential problem. After this a human can look and take a decision regarding whether this is to be expected or whether the system should be reversed to a previous version and /or the problem escalated." - "This is difficult [determining what scores are still expected], as to a certain extent you can make an estimate as to what is normal, but over time this can develop"	
	Simulated dataset on which the algorithm can be evaluated	-	- "have a simulated or a reassembled second set of data, where you actually know what the right outcomes are and keep running your model on that one as well. […] it is like a pilot in some sense, sometimes you put them in a simulator to see how they react in different situations."	
Algorithms cannot contain any bias	Control for bias should happen during development	-	- "Rather, you should do this [checking for bias] during the development of your model. Generally this [biases] are not things that change a lot over time, but are already known during the development of the model. I would think that something went wrong in the process if you find out about this at a later stage" - "Bias is unwanted, but classifying it as unexpected would be naïve. If your input data is biased it should not be unexpected"	
	Creation of a test dataset to flag potential bias	-	- - "A possible test that can be done to detect it [bias] is to create a dataset where you can control very explicitly whether the predictions have bias or not"	
	Measure using a fairness metric	(Arrieta et al., 2020)	- Imagine that you are predicting something, and you would expect the same percentage of positive predictions for both men and women, this is something you can just monitor. You can measure distribution of predicted values for both men and women, and these should be the same " - "The detection of unfairness or bias is not that difficult in principle. The monitoring techniques that are available work well for this. You only have to define the metric" - "according to GDPR in many cases bias cannot be measured. If you want to know whether you are discriminating minorities, you need data regarding who belongs to which minority. According to the GDPR you are not allowed to have this data and if you happen to have it by accident, you are not allowed to touch it" - "There are quite some different fairness metrics which exclude each other mathematically. It is impossible, literally impossible, mathematically impossible to make a model that is fair according to all these metrics. So, in this regard a choice has to be made which Is use-case specific"	
	Simple algorithms to infer the outcome of complex ones	-	- "If you train simple algorithms on your complex algorithms outcomes. And you can look in those simple models and you can see if there's a really big bias in your your your algorithm, you can see that."	
The systems utilising an	Saving information regarding model	(Miao et al., 2017;	- "The total tracking and versioning of data, models, environment, codes and experiments. That should all be tracked together"	

algorithm needs to be auditable	Saving information regarding historical performance metrics	Polyzotis et al., 2018; Schelter et al., 2017; Sridhar et al., 2018; Vartak et al., 2016)	- "So, what you can do for example is to version a dataset. You have saved the data on which a model was trained. This already gives information but requires that you can reconstruct what the data was at that moment [training the model / doing the prediction]. After this you can use version control to deploy a model and save the configuration. This will give you a good record of the model" - "You have to be able to connect all predictions to the model versions that were live, the data that was used as the input for the model, the model configuration, what data manipulation took place how the model was optimized" - "The concept lineage is very important to be able to trace every result - a model, individual prediction or metrics from experiments – to trace all this and see how things have come about. This is very close to reproducibility, but it is not the same. What you do is you save the entire heritage of a result, so for a prediction these are the settings, this was the data, these hyperparameters and this version of the code. This all helps you to reconstruct what actually happened" - "It is good practice to log data regarding who had access to the models and data. This allows, for example, insight into who had access to privacy-sensitive data." - "Lineage is not the same as reproducibility. To reproduce an individual prediction, you must trace back what happened to it. How the model was created and what data was used exactly. So, this is not just which dataset was used or the name of the database. Nor the columns that were used. The exact values for every field in the data need to be identical. This is not easy – partly due to the GDPR retention period. The systems that achieve this are complicated" - "To reproduce predictions, you need to be able to retrieve all data you had. I don't think this is GDPR-proof in some cases. Perhaps a certain time period can be used for which you can reproduce predictions"	
	Saving information regarding predictions			
	Saving information regarding model settings			
	Saving information regarding input data			
	Save information regarding all automatic and human changes made to models			
	Save code that created model and transformed data			
	Saving information regarding who has access			
	Knowledge needed to interpret and analyse predictions		- "You need data scientists that understand how the model works and what different metrics mean." - "You need someone, or have it documented somewhere, what the output of your model is and how it can be interpreted."	
Interpretability	Utilize inherently interpretable models	-	- "There are techniques to use interpretable models only. Inherently interpretable models, so that you don't need to create post hoc explanations for black-box models. And a lot of these type of models are not a lot worse performance wise. So, you can decide to use these if the use-case requires this"	
	Use of explainability algorithms		- "There are many tools to create explanations these days. They are also easy to use" - "It is quite difficult to do in a general manner [interpretability], as it depends on the shape of your data. The developer can choose an algorithm that fits for the use-case" - "it is quite easy to make an explanation, but you have no clue how good it is. So, you should do quality analysis of explanations" - "Perhaps you are familiar with the saying "All models are wrong, but some are useful". This is doubly true for explanations, as you start with a model, and all models are wrong, and to that you add another model [the explanation]. So, all explanations are wrong, but some are useful. Per use-case you must think about what you need from explanations and what kind of mistakes are acceptable" - "The way the explanations came to being involves many decisions. Different explainability algorithms might give different results"	
	Visualize predictions	(Kahng et al., 2016; Polyzotis et	- "You might also want to visualise predictions, as that might highlight important aspects"	

		al., 2018; Sculley et al., 2015)	- "Sometimes you also just want to visualise certain aspects of the algorithm"	
	Including model certainty in predictions	-	- "You can also include the certainty your model has regarding a prediction. This will show how confident the model is in this prediction. And perhaps you are not interested in just showing the confidence in this prediction, but the top five prediction the model was most confident about"	
	Knowledge of project needs to be documented to facilitate interpretability of results	-	- "You need data scientists that understand how the model works and what different metrics mean" - "You need someone, or have it documented somewhere, what the output of your model is and how it can be interpreted"	
Human-in-the-loop	Control predictions before use	(Baylor et al., 2017; Sridhar et al., 2018)	-" It can be that for every prediction someone has to take an action. This is seen in healthcare a lot, where the model predictions are used as additional information"	
	Control predictions after use	(Baylor et al., 2017; Sridhar et al., 2018)	- "You can control prediction after they are used and still take action if you think something went wrong"	
	Confirm model deployment	(Baylor et al., 2017; Sridhar et al., 2018)	- "You cannot control predictions, but control when a model gets deployed into production"	
	Control model deployment with a signoff that provides accountability	-	- "So, what you can have in git is that you have different branches and have a main branch on which the actual production environment is based. What you can do is that when you create a new version that a systems puts this live if it is accepted to the main branch. This main branch determines what is in production. What you don't want is that everyone can just make changes and put this to production, so you protect it. Someone needs to send a request to have his changes incorporated into the production environment and there are, for example, two senior people that need to approve these changes before they are incorporated. Something similar can be done for machine learning where any new model, or changes in the code, need to be reviewed by senior staff who gets information to decide before they are put to production. [..] this signoff can perhaps be seen as a digital signature"	
	Control based on alerts	(Baylor et al., 2017; Sridhar et al., 2018)	- "It can also be more reactive when you have a monitoring in place that alerts you if something occurred during the deployment of a new model"	